

Kitaplık Kodu :



**TÜRKİYE BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA
KURUMU
BİLİM İNSANI DESTEK PROGRAMLARI BAŞKANLIĞI**

**33. BİLİM OLİMPİYATLARI -2025
BİRİNCİ AŞAMA SINAVI**

ÖĞRENCİ - ÇÖZÜMLER

BİLGİSAYAR

Soru Kitaplığı Türü

A

17 Mayıs 2025 Cumartesi, 09.30 – 12.00

ADAYIN ADI SOYADI :
T.C. KİMLİK NO :
OKULU / SINIFI :
SINAVA GİRDİĞİ İL :

SINAVLA İLGİLİ UYARILAR:

- Bu sınav çoktan seçmeli 50 adet sorudan oluşmaktadır, süre 150 dakikadır.
- Her sorunun sadece bir doğru cevabı vardır. Doğru cevabınızı, cevap kâğıdınızdaki ilgili kutucuğu **tamamen karalayarak** işaretleyiniz. Soru kitapçığındaki hiç bir işaretleme değerlendirmeye alınmayacaktır.
- Her soru eşit değerinde olup, dört yanlış cevap bir doğru cevabı götürülecektir.** Boş bırakılan soruların değerlendirmede olumlu ya da olumsuz bir etkisi olmayacaktır.
- Sorular zorluk sırasında DEĞİLDİR. Dolayısıyla yanıtlamaya geçmeden önce bütün soruları gözden geçirmeniz önerilir.
- Sınavda herhangi bir yardımcı materyal, elektronik hesap makinesi ya da karalama kağıdı kullanılması yasaktır. Soru kitapçığındaki boşlukları karalama için kullanabilirsiniz.
- Sınav süresince görevlilerle konuşulması ve soru sorulması, öğrencilerin birbirlerinden kalem, silgi vb. şeyler istemeleri yasaktır.
- Sınav giren adayın bir soruya itiraz etmek istemesi durumunda, sınav soruları ve cevap anahtarı TÜBİTAK'ın internet sayfasında (<https://bilimolimpiyatlari.tubitak.gov.tr>) yayımlandıktan sonra 7 işgünü içerisinde, kanıtları ile birlikte, TÜBİTAK'a başvurusu gerekir. Bu tarihten sonra yapılacak başvurular işleme konmayacaktır. Sadece sınav giren adayın sorulara itiraz hakkı vardır, üçüncü kişilerin sınav sorularına itirazı işleme alınmayacaktır.
- TÜBİTAK Bilim Olimpiyatı-Birinci Aşama Sınavı'nda sorulan soruların üçüncü kişiler tarafından kullanılması sonucunda doğacak olan hukuki sorunlardan TÜBİTAK ve Atatürk Üniversitesi sorumlu tutulamaz. Atatürk Üniversitesi, bu tip durumlarda sorular ile ilgili görüş bildirmek zorunda değildir.
- Sınav sırasında kopya çeken, çekmeye teşebbüs eden ve kopya verenlerin kimlikleri sınav tutanağına yazılacak ve bu kişilerin sınavları geçersiz sayılacaktır. Görevliler kopya çekmeye veya vermeye kalkışanları uyararak sorumlu size aittir.
- Sınav başladıktan sonraki ilk yarım saat içinde sınav salonundan ayrılmak yasaktır.
- Sınav süresince sınav giriş belgenizi ve geçerli bir kimlik belgesini masanızın üzerinde bulundurunuz.
- Sınav salonundan ayrılmadan önce cevap kâğıdınızı ve soru kitapçığını görevlilere teslim etmeyi unutmayınız.

Başarılar dileriz.

A

BU SAYFA BOŞ BIRAKILMIŞTIR.

A

[Soru 1-3 için açıklama]

Farklı renklere sahip n adet zarınız var. Zarlarınız her bir yüzünde 1, 2, 3, 4, 5 ve 6 sayılarından birinin yer aldığı klasik altı yüzlü zarlardır. Size bir m tamsayısı veriliyor. n adet zarı attığınızda gelen n sayının toplamının m olmasının kaç farklı şekilde gerçekleşebileceğini hesaplamak istiyorsunuz.

Örnek: $n = 2$ adet zarınız varsa, toplamda $m = 3$ sayısını elde etmenin tam olarak iki farklı yolu vardır: zarların (1, 2) veya (2, 1) olarak gelmesi. Zarların farklı renklerde olduğuna ve bu nedenle bu iki durumun birbirinden farklı olduğuna dikkat ediniz.

SORU 1

$n=3$ adet zarınız olsun. $m=6$ sayısını kaç farklı şekilde elde edebilirsiniz?

- A) 7
- B) 8
- C) 9
- D) 10
- E) 11

ÇÖZÜM

Bu soru bütün durumlar sayılarak çözülebilir. Olası bütün durumlar:

(1, 1, 4), (1, 2, 3), (1, 3, 2), (1, 4, 1), (2, 1, 3), (2, 2, 2), (2, 3, 1), (3, 1, 2), (3, 2, 1) ve (4, 1, 1)'dir.

$m = 6$ sayısının toplam 10 şekilde elde edilebileceği görülmektedir. Doğru cevap D şıkkıdır.

CEVAP: D

A

SORU 2

$n=4$ adet zarınız olsun. $m=10$ sayısını kaç farklı şekilde elde edebilirsiniz?

- A) 43
- B) 56
- C) 68
- D) 74
- E) 80

ÇÖZÜM

Bu soru dinamik programlama kullanılarak çözülebilir. $Say(n, m)$ fonksiyonu n adet zar kullanılarak m sayısının kaç farklı şekilde elde edilebileceğini gösterebilir.

$n = 1$ ve $1 \leq m \leq 6$ olması durumunda $Say(n, m) = 1$ 'dir çünkü tek zar var ve bu zarın m gelmesi gerekmektedir.

$n = 1$ ve $(m \leq 0 \vee m \geq 7)$ olması durumunda $Say(n, m) = 0$ 'dir çünkü tek zar var ve bu zarın m gelmesi mümkün değildir.

Aksi takdirde $Say(n, m) = \sum_{t=1}^6 Say(n-1, m-t)$ 'dir. Dikkat edilirse bu özyineleme ifade

$1 \leq t \leq 6$ olmak üzere her t için son zarın t geldiği ve diğer zarların toplamının $m-t$ olduğu bütün durumların toplamını hesaplamaktadır.

Bu özyinelemeli ifade kullanılarak $1 \leq n \leq 4$ ve $1 \leq m \leq 10$ için $Say(n, m)$ değerleri hesaplandığında aşağıdaki M matrisindeki gibi olur.

$M = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 5 \ 4 \ 3 \ 0 \ 0 \ 1 \ 3 \ 6 \ 10 \ 15 \ 21 \ 25 \ 27 \ 0 \ 0 \ 0 \ 1 \ 4 \ 10 \ 20 \ 35 \ 56 \ 80)$

Doğru cevap E şıkında yer alan 80'dir.

CEVAP: E

SORU 3

n=20 adet zarınız olsun. m=25 sayısını kaç farklı şekilde elde edebilirsiniz?

- A) 8855
- B) 26334
- C) 42504
- D) 53130
- E) 74596

ÇÖZÜM

Bu soruyu bir önceki sorunun çözümünde verilen özyinelemeli ifade kullanılarak çözmek mümkün olsa bile çok fazla zaman alacaktır. Bu yüzden farklı bir çözüm gerekmektedir.

20 zarın herbiri en az 1 değerini alacaktır. Bu yüzden bizim yapmamız gereken kalan 5'i bu 20 zara kaç şekilde dağıtabileceğimizi bulmaktır. Bu problem 5 özdeş nesneyi ayırt edilebilir 20 kutuya yerleştirme problemidir ve çözümü $\left(\frac{5+20-1}{5}\right) = \left(\frac{24}{5}\right) = 42504$ 'tür.

CEVAP: C**SORU 4**

ARIBURNU kelimesindeki harfleri yer değiştirerek anlamlı ya da anlamsız 8 harfli kaç kelime oluşturulabilir?

- A) 2520
- B) 5040
- C) 10080
- D) 20160
- E) 40320

ÇÖZÜM

8 harf toplam 8! şekilde sıralanabilir. ARIBURNU kelimesinde U ve R harfleri ikişer defa geçmektedir. Bu harfler cevap $\frac{8!}{2!2!} = 10080$ 'dir.

CEVAP: C

A

SORU 5

ARIBURNU kelimesindeki harfleri yer deęiřtirerek oluřturulabilecek anlamlı ya da anlamsız 8 harfli kelimelerden bir řözlük oluřturuyorsunuz. Bu řözlükteki 1261. kelimenin ikinci harfi nedir? (Řözlükte yer alan tüm kelimeler alfabetik sıradadır.)

- A) A
- B) B
- C) N
- D) R
- E) U

ÇÖZÜM

Řözlükte en bařka A harfi ile bařlayan kelimeler yer alacaktır. Bařa A harfini yerleřtirirsek geri kalan 7 harften toplam $\frac{7!}{2!2!} = 1260$ kelime oluřturulabilir. Sıradaki kelime BAINRRUU’dur.

CEVAP: A

SORU 6

Camelot Krallığı'nda Arthur, Benedict, Caius, Dorian, Edric ve Fergus isimli 6 seçkin şövalye kralın düzenlediği yıllık onur töreninde 60 derece aralıklarla yuvarlak bir masaya oturacaktır. Ancak şövalyelerin oturma düzeni aşağıdaki 2 geleneksel kurala uymalıdır:

- Arthur'un yanına Benedict veya Caius oturamaz.
- Dorian ile Edric tam karşılıklı oturmalıdır.

Yuvarlak masa düzeninde dönüşümlü olarak aynı olan oturma şekilleri aynı kabul edilir. Kurallara uygun olacak şekilde şövalyeler masaya kaç farklı şekilde oturabilirler?

- A) 6
- B) 8
- C) 10
- D) 12
- E) 14

ÇÖZÜM

Yuvarlak masa etrafında dönüşlü simetri olduğu için bir kişiyi (örneğin Arthur'u) sabitleyebiliriz. Arthur'un soluna gelecek kişi 3 kişiden biri olabilir: Dorian, Edric, Fergus. Her bir durumu ayrı ayrı analiz edebiliriz.

- Arthur'un soluna Dorian otursun: Kurallara göre Edric, Dorian'ın tam karşısında oturur, Fergus da Arthur'un sağında yer alır. Kalan 2 sandalyeye Benedict ve Caius 2 şekilde oturabilir.
- Arthur'un soluna Edric otursun: Kurallara göre Dorian, Edric'in tam karşısında oturur, Fergus da Arthur'un sağında yer alır. Kalan 2 sandalyeye Benedict ve Caius 2 şekilde oturabilir.
- Arthur'un soluna Fergus otursun: Arthur'un sağına Dorian oturursa Edric de Dorian'ın karşısına oturur. Kalan 2 sandalyeye Benedict ve Caius 2 şekilde oturabilir. Arthur'un sağına Edric oturursa Dorian da Edric'in karşısına oturur. Kalan 2 sandalyeye Benedict ve Caius 2 şekilde oturabilir.

Yukarıdaki analizden görüleceği üzere süvariler masaya toplam 8 farklı şekilde oturabilirler.

CEVAP: B

SORU 7

Bir ailenin n tane çocuğu var. Anne ve baba, bayram sürprizi olarak her çocuğa bir hediye almış ve hediyeleri bayramda vermek üzere bir dolaba saklamış. Yaramaz çocuklar, bayramdan önce dolaptaki hediyeleri bulmuş ve her biri sırayla rastgele bir hediye seçip sahiplenmiş. Hiçbir çocuk, anne ve babasının kendisi için aldığı hediye sahiplenmemiş. Bu durumun gerçekleşme olasılığı $3/8$ olduğuna göre, ailenin kaç çocuğu vardır?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 7

ÇÖZÜM

Bize n tane birbirinden farklı nesne sıralı bir şekilde verilmiş olsun. Hiçbir nesnenin orijinal konumunda kalmadığı permütasyona verilen sıralı listenin bir düzensizliği denir. n elemanlı bir kümenin düzensizlik sayısı $D(n) = n! \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots + (-1)^n \frac{1}{n!} \right)$ eşitliğine göre hesaplanabilir. Bu soruda bizden $\frac{D(n)}{n!} = \frac{3}{8}$ eşitliğinin sağlanması için n 'nin alması gereken değer sorulmaktadır. $D(4) = 4! \left(1 - 1 + \frac{1}{2} - \frac{1}{6} + \frac{1}{24} \right) = 4! \times \frac{3}{8}$ olduğu için doğru cevap 4'tür.

CEVAP: B

A

SORU 8

Bir mühendis, değişik ömürlere sahip pilleri kullanarak bir cihazı mümkün olduğunca çok çalıştırmak istiyor. Cihaz sadece iki pil birlikte takıldığında çalışabiliyor ve her çalışmada cihaz, o iki pilin minimum ömrü kadar çalışıyor. Her çalıştırmada kullanılan iki pil tamamen tükeniyor.

$T = [10, 2, 9, 4, 7, 6, 7, 1, 5, 8, 4, 9, 6]$ dizisi pil ömürlerini saat cinsinden vermektedir. Mühendis cihazı toplamda en fazla kaç saat çalıştırabilir?

- A) 36
- B) 37
- C) 38
- D) 39
- E) 40

ÇÖZÜM

Bu sorunun optimal çözümü için uzun ömürlü pilleri birbiriyle kısa ömürlü pilleri de kendi aralarında eşleştirmek gerekir. Bu yüzden ilk olarak T dizisini büyükten küçüğe sıralayacağız. T dizisinin sıralı hali şu şekildedir. $T = [10, 9, 9, 8, 7, 7, 6, 6, 5, 4, 4, 2, 1]$.

İlk 2 pil 9 saat, sonraki 2 pil 8 saat, sonraki 2 pil 7 saat, sonraki 2 pil 6 saat, sonraki 2 pil 4 saat, sonraki 2 pil ise 2 saat boyunca cihazı çalıştırır. Sorunun cevabı $9 + 8 + 7 + 6 + 4 + 2 = 36$ eder.

CEVAP: A

A

[Soru 9-11 için açıklama]

Bir lisede TÜBİTAK araştırma projeleri kapsamında 8 öğrenci danışman öğretmenlerinin yönlendirmesiyle 3 ayrı gruba ayrılacaktır. Her öğrenci yalnızca bir gruba katılabilir ve hiçbir grup boş kalmamalıdır. Grupların isimlerinin ve öğrencilerin sıralamalarının bir önemi yoktur, yalnızca hangi öğrencilerin birlikte aynı grupta yer aldığını dikkate alıyoruz.

SORU 9

Bu 8 öğrenci kaç farklı şekilde 3 gruba ayrılabilir?

- A) 301
- B) 420
- C) 560
- D) 966
- E) 1094

ÇÖZÜM

Stirling sayılarının ikinci türü $S(n, k)$, n elemanlı bir kümenin k tane boş olmayan, sırasız alt kümeye bölünme sayısını verir. Hesaplama aşağıdaki özyinelemeli ifade kullanılarak yapılabilir.

- $S(0, 0) = 1$
- $n > 0$ için $S(n, 0) = 0$
- $k > 0$ için $S(0, k) = 0$
- Diğer durumlar için $S(n, k) = k \times S(n - 1, k) + S(n - 1, k - 1)$ özyinelemesine göre yapılabilir.

Bu formülün neden doğru olduğunu anlamak için n elemanlı bir kümede son elemanı ele alalım. Bu eleman 2 farklı şekilde yerleştirilebilir:

1. Bu eleman mevcut k gruptan birine konur. Bu durumda geriye kalan $n - 1$ eleman zaten k gruba ayrılmıştır. Bu yerleştirme sayısı $S(n - 1, k)$ ile gösterilmektedir. Eleman k gruptan herhangi birine yerleştirilebileceği için bu yerleştirme $k \times S(n - 1, k)$ şekilde yapılabilir.
2. Bu eleman tek başına bir grup oluşturur. Bunun olması için önceki $n - 1$ elemanın $k - 1$ gruba ayrılmış olması gerekmektedir. Bu da $S(n - 1, k - 1)$ şekilde yapılabilir.

Yukarıda verilen özyineleme kullanılarak $0 \leq n \leq 7$ ve $0 \leq k \leq 3$ için $S(n, k)$ değerleri hesaplandığında aşağıdaki M matrisi elde edilir.

$M = (1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 3 \ 1 \ 0 \ 1 \ 7 \ 6 \ 0 \ 1 \ 15 \ 25 \ 0 \ 1 \ 31 \ 90 \ 0 \ 1 \ 63 \ 301 \ 0 \ 1 \ 127 \ 966)$
Doğru cevap D şıkkındaki 966 olacaktır.

CEVAP: D

SORU 10

Araştırma projesi yapacak öğrencilerden Asu ve Ulus arasında tartışma çıkmış ve aynı grupta yer almak istemiyorlar. Asu ve Ulus farklı gruplarda olacak şekilde bu 8 öğrenci kaç farklı şekilde 3 gruba ayrılabilir?

- A) 280
- B) 483
- C) 547
- D) 665
- E) 720

ÇÖZÜM

Bir önceki sorunun çözümünden bu 8 öğrencinin 966 şekilde 3 gruba ayrılabilildiğini biliyoruz. İlk olarak bu 966 gruptan kaç tanesinde Asu ve Ulus'un aynı grupta yer aldığını bulalım. Bunun için Asu ve Ulus'u tek bir kişi olarak düşünüp 7 öğrencinin kaç farklı şekilde 3 gruba ayrılabilceğini bulabiliriz. Bir önceki sorunun çözümünden bunun $S(7, 3) = 301$ olduğunu biliyoruz. 966 gruptanın 301 tanesinde Asu ve Ulus aynı grupta olduğuna göre $966 - 301 = 665$ tanesinde farklı grupta olurlar.

CEVAP: D

SORU 11

Asu ve Ulus barışmışlar ve çok iyi iki arkadaş olmuşlar. Öğretmenlerine mutlaka aynı grupta yer almak istediklerini söylemişler. Ancak bu sefer de Canan ve Deniz arasında tartışma çıkmış ve kesinlikle aynı grupta yer almak istemiyorlarmış. Asu ve Ulus'un aynı grupta, Canan ve Deniz'in ise farklı grupta yer almaları şartlarını sağlayacak şekilde bu 8 öğrenciyi kaç farklı şekilde 3 gruba ayırabiliriz?

- A) 90
- B) 118
- C) 147
- D) 185
- E) 211

ÇÖZÜM

Bir önceki sorunun çözümünde bu 8 öğrencinin Asu ve Ulus aynı grupta yer alacak şekilde $S(7, 3) = 301$ şekilde 3 gruba ayrılabilildiğini biliyoruz. İlk olarak bu 301 gruptan kaç tanesinde Canan ve Deniz'in aynı grupta yer aldığını bulalım. Bunun için Asu ve Ulus'u tek bir kişi olarak düşündüğümüz gibi Canan ve Deniz'i de tek bir kişi gibi düşünüp 6 öğrencinin kaç farklı şekilde 3 gruba ayrılabilceğini bulabiliriz. İki önceki sorunun çözümünde oluşturduğumuz M matrisinden bunun $S(6, 3) = 90$ olduğunu biliyoruz. Asu ve Ulus'un aynı grupta yer aldığı 301 gruptan 90 tanesinde Canan ve Deniz aynı grupta olduğuna göre $301 - 90 = 211$ tanesinde farklı grupta olurlar.

CEVAP: E

A

[Soru 12-13 için açıklama]

Bir dağ yolunda, n tane manzara noktası yol boyunca sıralı bir şekilde bulunuyor. Her bir manzara noktasında, i numaralı noktada çekilen fotoğrafın güzellik değeri $T[i]$ olarak veriliyor ve $T[i]$ pozitif bir tam sayıdır. Ancak, sınırlı zamanınız olduğu için yan yana iki manzara noktasında durarak fotoğraf çekemezsiniz. Amacınız, seçtiğiniz manzara noktalarından çekilen fotoğrafların toplam güzellik değerini mümkün olduğunca büyük yapmaktır.

Örnek: Eğer $T = [10, 5, 15, 25, 3, 8]$ olsaydı, maksimum toplam güzellik değeri $10 + 25 + 8 = 43$ olurdu.

SORU 12

$T = [12, 18, 7, 35, 4, 20, 28, 9, 14, 22]$ için maksimum toplam güzellik değeri ne olur?

- A) 101
- B) 104
- C) 107
- D) 109
- E) 112

ÇÖZÜM

Bu problem dinamik programlama ile çözülebilir. Bir DP dizisi tanımlayalım: $DP[i]$, ilk i manzara noktasına kadar (i dahil) elde edilebilecek maksimum güzellik değerini temsil eder. Her i . noktada aşağıdaki iki seçeneğin olduğuna dikkat ediniz:

- i . noktayı seç: Bu durumda $(i - 1)$. noktayı seçemeyiz, bu yüzden maksimum güzellik değeri $T[i] + DP[i - 2]$ 'dir.
- i . noktayı seçme: Bu durumda maksimum güzellik değeri $DP[i - 1]$ 'dir.

Bu yüzden DP dizisinin elemanlarını aşağıdaki özyinelemeli ifadeye göre hesaplayabiliriz.

$$DP[0] = T[0], \quad DP[1] = \max(T[0], T[1]), \quad \text{ve} \quad i \geq 2 \quad \text{için} \\ DP[i] = \max(DP[i - 1], T[i] + DP[i - 2]).$$

Soruda verilen T dizisi için DP dizisi: $DP = [12, 18, 19, 53, 53, 73, 81, 82, 95, 104]$ olur. Bu yüzden doğru cevap 104 olur.

CEVAP: B

SORU 13

$T = [15, 30, 10, 45, 20, 35, 25, 50, 5, 40, 15, 60, 30, 10, 55, 20, 25, 5, 15, 5]$ için maksimum toplam güzellik değeri ne olur?

- A) 345
- B) 350
- C) 355
- D) 360
- E) 365

ÇÖZÜM

Bu problemin çözümü bir önceki sorunun çözüm kısmında detaylıca açıklanmıştı.

Soruda verilen T dizisi için DP dizisi aşağıdadır.

$DP = [15, 30, 30, 75, 75, 110, 110, 160, 160, 200, 200, 260, 260, 270, 315, 315, 340, 340, 355,$

Bu yüzden doğru cevap 355 olur.

CEVAP: C

SORU 14

Mantık Kasabası'nda yaşayan insanlar ya doğrucu (her zaman doğruyu söyleyen) ya da yalancı (her zaman yalan söyleyen) olur. Bir gün bu kasabada yaşayan Ayşe, Burak ve Cem arasında şu konuşmalar geçiyor:

- Ayşe: "Burak yalancıdır."
- Burak: "Cem yalancıdır."
- Cem: "Ben ve Burak'tan en az birimiz doğrucudur."

Bu ifadeler ışığında, aşağıdakilerden hangisi doğrudur?

- A) Ayşe ve Cem doğrucudur.
- B) Sadece Ayşe doğrucudur.
- C) Sadece Burak doğrucudur.
- D) Sadece Cem doğrucudur.
- E) Burak ve Cem doğrucudur.

ÇÖZÜM

Cem'in yalancı olduğunu varsayalım. O zaman "Ben ve Burak'tan en az birimiz doğrucudur." ifadesi yanlıştır. Yani hem Cem hem Burak yalancıdır. Burak yalancı olduğu için "Cem yalancıdır" ifadesi yanlıştır yani Cem doğrucudur. Bu da baştaki varsayımızla çelişmektedir. Öyleyse Cem doğrucudur.

Cem doğrucudur. Bu yüzden Burak'ın "Cem yalancıdır" ifadesi yanlıştır ve dolayısıyla Burak yalancıdır. Ayşe'nin "Burak yalancıdır" ifadesi doğru olduğu için Ayşe doğrucudur.

CEVAP: A

SORU 15

Dört arkadaş (Ayşe, Burak, Cem ve Deniz) bir yaz kampına katılıyor. Her biri farklı bir spor dalında uzmanlaşıyor: yüzme, koşu, bisiklet ve tenis. Ayrıca, her biri farklı bir renkte çadırda kalıyor: mavi, kırmızı, yeşil ve sarı. Kimin hangi spor dalında uzmanlaştığına ve hangi renk çadırda kaldığına ilişkin olarak size aşağıdaki bilgiler verilmiştir.

1. Ayşe, mavi çadırda kalmıyor.
2. Yüzme uzmanı, kırmızı çadırda kalıyor.
3. Burak, tenis uzmanı değil.
4. Cem, yeşil çadırda kalıyor.
5. Tenis uzmanı, Ayşe değil.
6. Bisiklet uzmanı, sarı çadırda kalıyor.
7. Deniz, koşu uzmanı değil.

Burak hangi renk çadırda kalmaktadır?

- A) Mavi
- B) Kırmızı
- C) Yeşil
- D) Sarı
- E) Burak'ın hangi renk çadırda kaldığı belirlenemez

ÇÖZÜM

Verilen ipuçları Burak'ın hangi renk çadırda kaldığını belirlemek için yeterli değildir. Bunu görmek için ipuçları ile uyumlu aşağıdaki iki çözüme bakmak yeterlidir.

Çözüm 1: Yüzme uzmanı Deniz kırmızı çadırda, koşu uzmanı Burak mavi çadırda, bisiklet uzmanı Ayşe sarı çadırda, tenis uzmanı Cem yeşil çadırda kalmaktadır. İpuçları ile uyumlu bu çözüme göre Burak mavi çadırda kalmaktadır.

Çözüm 2: Yüzme uzmanı Ayşe kırmızı çadırda, koşu uzmanı Cem yeşil çadırda, bisiklet uzmanı Burak sarı çadırda, tenis uzmanı Deniz mavi çadırda kalmaktadır. İpuçları ile uyumlu bu çözüme göre Burak sarı çadırda kalmaktadır.

CEVAP: E

A

[Soru 16-17 için açıklama]

Herbiri bir anne, bir baba ve iki çocuktan oluşan 3 aile (A, B ve C aileleri) beraber bir oyun oynayacaklardır. Oyun gereği, üçer kişilik 4 takım kurulacaktır. Her çocuk mutlaka en az bir ebeveyni ile aynı takımda olmak zorundadır.

SORU 16

A ailesinin babasının olduğu takımda A ailesinden hiç çocuk yoksa, aşağıdakilerden hangisi doğrudur?

- A) B ve C ailelerinden birisinin iki çocuğu da aynı takımdadır.
- B) B ve C ailelerinden birer çocuk aynı takımdadır.
- C) A ailesinin annesi ile B veya C ailelerinden birisinin annesi aynı takımdadır.
- D) Bu kuralları sağlayan 16 farklı gruplama yapılabilir.
- E) Diğer şıkların tümü yanlıştır.

ÇÖZÜM

A1 baba, A2 anne, A3 ve A4 çocuklar olacak şekilde kodlayalım. Hiçbir aileden hem anne hem de baba aynı grupta olamaz. Aksi takdirde bu ailenin bir çocuğu ebeveynlerinden farklı bir grupta olmak zorunda kalır. A1 in grubunda B1, B2, C1, C2 den birisi olmak zorundadır, bu kişiye X diyelim. X'in kim olduğu 4 farklı şekilde belirlenebilir. Bundan sonra iki durum söz konusudur.

1. Bu gruptaki 3. kişi de B1, B2, C1, C2 arasından gelir, buna da Y diyelim. Ancak bu durumda X ve Y farklı aileler olmak zorundadır. Bu da $4 \times 2 = 8$ ihtimal yapar. Ancak tekrarlardan dolayı $8/2 = 4$ ihtimale düşer.

2. Gruptaki 3. kişi X in çocuklarından biridir. Bu da $4 \times 2 = 8$ ihtimal yapar.

Birinci durumda, kalan grupların her biri bir ebeveyn ve onun çocuklarından oluşmak zorundadır. İkinci durumda ise, A2 kendi çocuklarıyla bir grup oluşturur, X in diğer ebeveyni ve diğer çocuğu Y ailesinden bir ebeveynle bir grup oluşturur ki burada 2 ihtimal vardır. Ve son grup da Y nin diğer ebeveyni ve 2 çocuğundan oluşur.

Tüm ihtimaller değerlendirildiğinde (C) şıkkının doğru olduğu görülür.

CEVAP: C

SORU 17

A ailesinin babası iki çocuğuyla aynı gruptaysa diğer gruplar kaç farklı şekilde belirlenebilir?

- A) 8
- B) 12
- C) 16
- D) 20
- E) 24

ÇÖZÜM

A1 iki çocuğuyla aynı grupta ise, bu durumda A2 için iki durum vardır.

1. B ve C ailelerinden birer ebeveynle aynı gruptadır, ki bu $2 \times 2 = 4$ ihtimal eder. Bu durumda diğer iki grup B ve C ailelerindeki diğer ebeveynler ve onların çocukları olacak şekilde oluşmak zorundadır.

2. A2, B veya C ailelerinden bir ebeveyn ve onun bir çocuğu ile aynı gruptadır. Bu $4 \times 2 = 8$ ihtimal eder. Diyelim ki bu X ailesinden bir ebeveyn ve çocuğu olsun. X ailesindeki diğer ebeveyn ve diğer çocuk artık bir grupta olmak zorundadır. Bu 3. gruptaki 3. kişi de Y ailesinden bir ebeveyn olmak zorundadır. Bu da iki ihtimal eder. Yani toplamda $8 \times 2 = 16$ ihtimal söz konusu olur.

Yani genel toplamda 20 ihtimal söz konusudur.

CEVAP: D

A

SORU 18

- 1- Hava güzel olduğunda bisiklete binerim.
- 2- Bugün bisiklete binmedim, ama film izledim.
- 3- Hava güzel değilse ve film izlediysem, o gün pizza yerim.
- 4- Bugün ya pizza yemedim veya lahmacun yemedim.

Yukarıdaki dört önermenin de doğru olduğu bilindiğine göre, aşağıdakilerden hangisi çıkarılamaz?

- A) Bugün hava güzel değildi.
- B) Film izledim ve lahmacun yemedim.
- C) Lahmacun yediysem film izlemişimdir.
- D) Hava güzeldir veya film izlemişimdir.
- E) Bisiklete binmediysem lahmacun yemişimdir.

ÇÖZÜM

Aşağıdaki önermeleri tanımlayalım:

p: hava güzel, q: bisiklete bindim, r: film izledim, s: pizza yedim, t: lahmacun yedim

Verilen doğrular bu durumda şu şekilde ifade edilebilir:

- 1- p ise q,
- 2- q değil ve r,
- 3- p değil ve r ise s,
- 4- s değil veya t değil.

Bunlara bağlı olarak

- 5- q değil ise p değil , 1'in kontrapozisyonu
- 6- q değil , 2'den direk elde edilir
- 7- p değil , 5 ve 6 ya Modus Ponens uygulanır
- 8- r , 2'den direk elde edilir
- 9- p değil ve r , 7 ve 8 den
- 10- s , 3 ve 9 dan Modus Ponens ile
- 11- t değil , 4 ve 10 dan elde edilir.

Şıklara baktığımızda,

- A) "p değil" , 7 den çıkarılabilir
- B) "r ve t değil" , 8 ve 11 den çıkarılabilir
- C) "t ise r" , t yanlış olduğundan her zaman doğrudur
- D) "p veya r" , 8 den dolayı doğrudur.
- E) "q değil ise t" , yanlıştır çıkarılamaz.

CEVAP: E

SORU 19

10 erkek ve 5 kadından oluşan bir mecliste 5 kişilik bir komisyon oluşturulacaktır. Komisyondaki kadın üyelerin sayısı erkek üyelerden fazla olmak zorunda ve en az bir kadın ve bir erkek olmak zorunda ise, bu komisyon kaç farklı şekilde oluşturulabilir?

- A) 525
- B) 500
- C) 475
- D) 450
- E) 425

ÇÖZÜM

$$(5,4)(10,1)+(5,3)(10,2) = 500$$

CEVAP: B

SORU 20

Bir torbada 9 adet özdeş görünümlü bilye bulunmaktadır. Bu bilyelerden 7 tanesi ağırlık olarak da özdeş olsa da, diğer 2 bilye birbirine özdeş ağırlıkta ama diğer 7 tanesinden daha hafiftir. Elinde yalnızca çift kefeli bir terazi bulunan bir öğrenci daha hafif olan iki bilyeyi en az tartımda bulmak istiyor. Öğrencinin olabilecek en doğru algoritmayla bu bilyeleri bulduğunu kabul edersek, en kötü durumda kaç tartım yapmış olabilir.

- A) 2
- B) 3
- C) 4
- D) 5
- E) 6

ÇÖZÜM

Bilyeleri üçerli 3 gruba ayıralım. Bu gruplara A,B,C isimlerini verelim. Bilyeler de A1,A2,A3 gibi isimlendirilmiş olsun.

– Tartım 1: A ve B grupları tartılır. İki ihtimal vardır:

- A ve B eşit gelir.

Tartım 2: B ve C tartılır.

- B ve C eşit gelemez, aksi takdirde A, B ve C nin hepsi eşit olur.
- B hafif gelir. Bu durumda A ve B de birer hafif top vardır.

Tartım 3: A1 ve A2 tartılır. Biri hafif gelirse hafif top bulunmuş olur.

Eşit gelirlirse A3 hafif olarak bulunmuş olur.

Tartım 4: B1 ve B2 tartılır. Biri hafif gelirse hafif top bulunmuş olur.

Eşit gelirlirse B3 hafif olarak bulunmuş olur.

- C hafif gelir. Yukarıdakine benzer şekilde devam edilir.

- A veya B den birisi hafif gelir. A'nın hafif geldiğini varsayalım. B'nin hafif geldiği durum özdeşdir.

Tartım 2: B ve C tartılır.

- B ve C eşit gelir. Bu durumda hafif iki top A'dadır.

Tartım 3: A1 ve A2 tartılır. Eşitlerse ikisi de hafiftir. Birisi hafif gelirse o top ve A3 hafiftir.

- C hafif gelir.

Tartım 3: A1 ve A2 tartılır. Biri hafif gelirse hafif top bulunmuş olur.

Eşit gelirlirse A3 hafif olarak bulunmuş olur.

Tartım 4: C1 ve C2 tartılır. Biri hafif gelirse hafif top bulunmuş olur.

Eşit gelirlirse C3 hafif olarak bulunmuş olur.

- A, B'den hafif olduğundan B, C'den hafif olamaz.

CEVAP: C

SORU 21

İki çocuk belirli bir adım mesafesinde duruyor ve sırayla birbirlerine doğru adım atıyorlar. Sırası gelen çocuk diğerine doğru 1, 2 veya 3 adım atıyor. Son adımı atıp aradaki mesafeyi sıfırlayan çocuk oyunu kazanıyor. Her iki çocuğun da her zaman olabilecek en iyi hamleleri yaptıkları kabul edilirse, aşağıdaki mesafelerden hangisinde oyuna başlarsa oyuna başlayan çocuk her zaman oyunu kaybeder?

- A) 64
- B) 73
- C) 82
- D) 90
- E) 111

ÇÖZÜM

Tersten yaklaşırsak, aradaki mesafe 1,2 veya 3 adım olduğunda sıradaki oyuncu oyunu kazanabilir. 4 adım olduğunda ise yapacağı hamle ile mesafeyi 1,2 veya 3'e düşürmek zorundadır ve bu da rakibini oyunu kazanacağı bir pozisyona sokmaya mecbur olduğu anlamına gelir. Yani 4 birim mesafe sıradaki oyuncunun kaybedeceği anlamına gelir. 5 birim mesafede sırası gelen oyuncu 1 adım atarak rakibini 4 birim mesafe pozisyonuna, yani kaybetme pozisyonuna sokabilir. Benzer şekilde 6 birimde sırası gelen oyuncu 2 adım, 7 birimde sırası gelen oyuncu ise 3 adım atarak rakibini kaybetme pozisyonuna sokar. Yani 5,6 ve 7 adım kazanma pozisyonlarıdır. Benzer şekilde 8 adım ise kaybetme pozisyonudur. Görüldüğü üzere her 4 adımda bir kaybetme pozisyonu gelir, diğer durumlar ise kazanma pozisyonlarıdır. Dolayısıyla 4'ün katları sıradaki oyuncu için kaybetme pozisyonudur. Şıkların içinde 4'ün katı olan 64 sıradaki oyuncu için yani oyuna başlayan oyuncu için kaybetme pozisyonudur ve oyunu her zaman 2. oyuncu kazanır.

CEVAP: C

SORU 22

Bir düzlemin üzerine çizilen, herhangi ikisi paralel olmayan ve herhangi üçü aynı noktadan geçmeyen 5 doğru düzlemi kaç bölgeye ayırır?

- A) 10
- B) 25
- C) 15
- D) 16
- E) 12

ÇÖZÜM

Verilen kısıtlar gereği her doğru diğer doğruların tümünü kesmek durumundadır. Tek tek doğruları düzleme eklersek,

Boş düzlem tek bölgedir.

1. doğru düzlemi iki bölgeye ayırır
2. doğru bu iki bölgeden de geçer ve düzlemi 4 bölgeye ayırır
3. doğru var olan 2 doğruyu keserken 3 bölgeden geçer ve düzlemi $4+3=7$ bölgeye ayırır.
4. doğru var olan 3 doğruyu keserken 4 bölgeden geçer ve düzlemi $7+4=11$ bölgeye ayırır.
5. doğru var olan 4 doğruyu keserken 5 bölgeden geçer ve düzlemi $11+5=16$ bölgeye ayırır.

CEVAP: D

A

SORU 23

Aşağıdaki algoritma verilen A ve B isminde iki adet ikili diziyi kullanarak ekrana “Yukarı” ve “Sağ” komutlarını yazdırmaktadır. Bu komutlar ise yazdırıldıkları sırada bir robota aktarılmakta ve robotun bir odada engellere çarpmadan hedefe gitmesini sağlamaktadır.

1. Mevcut dizi olarak A dizisini belirle
2. Mevcut dizide okunacak değer olduğu sürece
 - 2.1. Mevcut dizide sıradaki ikili değeri oku
 - 2.2. Okunan değer 1 ise
 - 2.2.1. Mevcut dizi A ise ekrana “Yukarı” yazdır
 - 2.2.2. Mevcut dizi B ise ekrana “Sağ” yazdır
 - 2.3. Okunan değer 0 ise
 - 2.3.1. Mevcut dizi A ise mevcut diziyi B yap
 - 2.3.2. Mevcut dizi B ise mevcut diziyi A yap

Robotun içinde bulunduğu odanın şeması aşağıda verilmiştir. Robot R harfi ile, engeller E harfi ile, hedef ise H harfi ile gösterilmiştir:

		E		H
E				
			E	
	E			
R			E	

Aşağıda verilen A ve B dizilerinden hangisi bu odada robotun hedefe ulaşmasını sağlayacaktır?

- A) $A=\{1, 1, 0, 0, 1, 1, 0\}$ $B=\{0, 1, 1, 0, 1, 1\}$
B) $A=\{0, 0, 1, 1, 1, 0, 0, 1, 0\}$ $B=\{1, 0, 1, 0, 0, 1, 0, 1\}$
C) $A=\{1, 1, 0, 1, 0, 1\}$ $B=\{1, 1, 0, 1, 1\}$
D) $A=\{0, 1, 1, 1, 0, 1, 0\}$ $B=\{1, 1, 0, 1, 0\}$
E) $A=\{1, 0, 1, 0, 1, 0, 0, 1\}$ $B=\{0, 1, 0, 1, 1, 1, 0\}$

ÇÖZÜM

Y: Yukarı, S: Sağ, X: Değiştir

- A) Y, Y, X, X, X, S, S, X, Y, Y -> Engele çarptı
B) X, S, X, X, S, X, Y, Y, Y, X, X, X, S, X, Y, X, S -> Hedefe ulaştı
C) Y, Y, X, S, S, X, Y, X, S, S -> Hedefe ulaşamadan durdu
D) X, S, S, X, Y, Y, Y, X, S, X, Y, X -> Hedefe ulaşamadan durdu
E) Y, X, X, Y, X, S, X, Y, X, S, S, S, X, X -> Hedefe ulaşamadan durdu

CEVAP: B

A

A

SORU 24

```
ALGORITMA GizemliIslem(Dizi A, tamsayı n, tamsayı k)
  YeniDizi ← boş bir dizi
  YeniBoyut ← 0
  for i ← 0 to n-1 do
    sayac ← 0
    for j ← 0 to YeniBoyut-1 do
      if A[i] == YeniDizi[j] then
        sayac ← sayac + 1
      end if
    end for
    if sayac < k then
      YeniDizi[YeniBoyut] ← A[i]
      YeniBoyut ← YeniBoyut + 1
    end if
  end for
  return YeniDizi
```

Yukarıdaki GizemliIslem algoritması, $A = [1, 2, 2, 1, 3, 2]$ dizisi, $n=6$ ve $k = 2$ değeri ile çağrıldığında, dönen (return edilen) YeniDizi'nin içeriği ne olur? ($\leftarrow$ atama işlemidir.)

- A) [1, 2, 3]
- B) [1, 2, 3, 2]
- C) [2, 1, 3]
- D) [1, 2, 2, 1, 3]
- E) Hiçbiri

ÇÖZÜM

Algoritma, girdi dizisi **A**'yı tarayarak **YeniDizi** adında yeni bir dizi oluşturur. Bir elemanın **YeniDizi**'ye eklenip eklenmeyeceği, o elemanın **YeniDizi**'de kaç kez tekrar ettiğine bağlıdır. Bir eleman, **YeniDizi**'de k 'dan daha az sayıda bulunuyorsa, **YeniDizi**'ye eklenir. $k=2$ olduğu için, her elemandan en fazla 1 tane **YeniDizi**'de yer alabilir (ilk tekrar hariç).

Adım adım $A = [1, 2, 2, 1, 3, 2]$ ve $k = 2$ için algoritmayı izleyelim:

1. $i = 0, A[0] = 1$. **YeniDizi** = []. $sayac = 0 < 2$. **YeniDizi** = [1], **YeniBoyut** = 1.
2. $i = 1, A[1] = 2$. **YeniDizi** = [1]. $sayac = 0 < 2$. **YeniDizi** = [1, 2], **YeniBoyut** = 2.

A

3. $i = 2, A[2] = 2$. $YeniDizi = [1, 2]$. $sayac = 1 < 2$. $YeniDizi = [1, 2, 2]$, $YeniBoyut = 3$.
4. $i = 3, A[3] = 1$. $YeniDizi = [1, 2, 2]$. $sayac = 1 < 2$. $YeniDizi = [1, 2, 2, 1]$, $YeniBoyut = 4$.
5. $i = 4, A[4] = 3$. $YeniDizi = [1, 2, 2, 1]$. $sayac = 0 < 2$.
 $YeniDizi = [1, 2, 2, 1, 3]$, $YeniBoyut = 5$.
6. $i = 5, A[5] = 2$. $YeniDizi = [1, 2, 2, 1, 3]$. $sayac = 2 < 2$
(yanlış). $YeniDizi$ değişmez.

Sonuç olarak, döndürülen $YeniDizi [1, 2, 2, 1, 3]$ olur.

CEVAP: D

A

[Soru 25-26 için açıklama]

Bir algoritma, girdi olarak pozitif tam sayılardan oluşan bir dizi ve bir eşik değeri T alır. Algoritma, dizideki her bir başlangıç noktası için, bu noktadan başlayarak oluşturulabilecek tüm ardışık alt dizilerin toplamlarını hesaplar. Eşik değerini aşan ilk alt dizinin başlangıç ve bitiş indislerini bir çift olarak saklar. Her başlangıç noktası için en fazla bir indis çifti saklanır. Algoritma, tüm olası başlangıç noktalarını denedikten sonra sakladığı indis çiftlerinin listesini döner.

SORU 25

Algoritma $A = [1, 4, 2, 3, 5]$ dizisi ve $T = 6$ eşik değeri ile çalıştırıldığında, algoritmanın döneceği kaç indis çifti vardır?

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

ÇÖZÜM

1. **Başlangıç 0 (1):** $[1, 4, 2]$ (toplam $7 > 6$). İndisler (0, 2).
2. **Başlangıç 1 (4):** $[4, 2, 3]$ (toplam $9 > 6$). İndisler (1, 3).
3. **Başlangıç 2 (2):** $[2, 3, 5]$ (toplam $10 > 6$). İndisler (2, 4).
4. **Başlangıç 3 (3):** $[3, 5]$ (toplam $8 > 6$). İndisler (3, 4).
5. **Başlangıç 4 (5):** Yok.

Döndürülen liste: $[(0, 2), (1, 3), (2, 4), (3, 4)]$.

Yani toplam 4 tane çift vardır.

CEVAP: D

SORU 26

Algoritma, $A = [2, 1, 5, 1, 3, 1]$ dizisi ve $T = 10$ eşik değeri ile çalıştırıldığında, algoritmanın döneceği (return edeceği) kaç indis çifti vardır?

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

ÇÖZÜM**1. Başlangıç İndisi 0 (Sayı 2):**

- $[2]$ (toplam 2) < 10
- $[2, 1]$ (toplam 3) < 10
- $[2, 1, 5]$ (toplam 8) < 10
- $[2, 1, 5, 1]$ (toplam 9) < 10
- $[2, 1, 5, 1, 3]$ (toplam 12) > 10 . İlk aşan bulundu. İndis çifti: $(0, 4)$ saklanır.

2. Başlangıç İndisi 1 (Sayı 1):

- $[1]$ (toplam 1) < 10
- $[1, 5]$ (toplam 6) < 10
- $[1, 5, 1]$ (toplam 7) < 10
- $[1, 5, 1, 3]$ (toplam 10) $= 10$ (aşmıyor)
- $[1, 5, 1, 3, 1]$ (toplam 11) > 10 . İlk aşan bulundu. İndis çifti: $(1, 5)$ saklanır.

3. Başlangıç İndisi 2 (Sayı 5):

- $[5]$ (toplam 5) < 10
- $[5, 1]$ (toplam 6) < 10
- $[5, 1, 3]$ (toplam 9) < 10
- $[5, 1, 3, 1]$ (toplam 10) $= 10$ (aşmıyor)

4. Başlangıç İndisi 3 (Sayı 1):

- $[1]$ (toplam 1) < 10
- $[1, 3]$ (toplam 4) < 10
- $[1, 3, 1]$ (toplam 5) < 10

A

5. Başlangıç İndisi 4 (Sayı 3):

- $[3]$ (toplam 3) < 10
- $[3, 1]$ (toplam 4) < 10

6. Başlangıç İndisi 5 (Sayı 1):

- $[1]$ (toplam 1) < 10

Saklanan indis çiftleri listesi: $[(0, 4), (1, 5)]$.

Bu durumda, algoritma toplamda 2 indis çifti döndürür.

CEVAP: B

SORU 27

Bir yapıda, 1'den n 'e kadar etiketlenmiş n farklı konum (odalar, koridorlar, balkonlar vb.) vardır. Ayrıca, k tane (a, b) çifti verilmiştir ve bir çift, a konumunun b konumuna doğrudan kapı ile bağlı olduğunu ve anahtara sahip olduğunuzu gösterir. Giriş konumu 1, balkon ise n olarak etiketlenmiştir. Girişten balkona gidip gidemeyeceğinizi bulan bir kod yazarken aşağıdakilerden hangisi dikkat etmemiz gereken (anlamalı) bir husus değildir?

- A) Konumlar arası bağlantılar kaydedilirken uygun bir veri yapısı kullanmak.
- B) Bağlı konumlar için hem (a, b) hem (b, a) olacak şekilde çift yönlü bağlantılar kaydetmek.
- C) (a, b) çiftlerini $a < b$ olacak şekilde sıralayıp, 1 ve n konumlarının ardışık olup olmadığına bakmak.
- D) Ziyaret edilen konumları tutarak aynı konuma tekrar girip sonsuz döngüye düşmeyi engellemek.
- E) Kodun, balkonu (konum n) bulana kadar her yeni komşuyu sırayla incelemesini sağlamak.

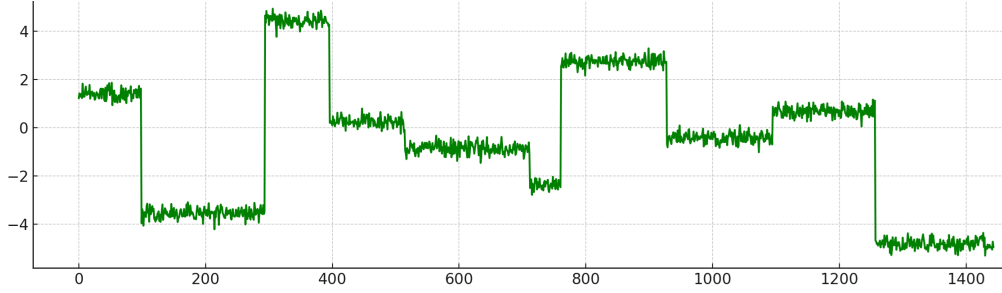
ÇÖZÜM

Bu beş maddeden dört tanesi (A, B, D, E) gerçekten girişten balkona varış algoritmasında dikkat edilmesi gereken veya işe yarayan hususlardır. C seçeneğinde sadece çiftleri sıralayıp 1 ve n konumlarının yanyana oluşuna bakmak, çok adımlı (dolaylı) bağlantıları tamamen göz ardı ettiği için soruda belirtilen “gidip gidememe” durumunu güvenilir biçimde tespit etmeyecektir. Bu nedenle dikkat edilmesi gereken (anlamalı) bir husus değildir. Doğru cevap C şıkkıdır.

CEVAP: C

SORU 28

Aşağıdaki gürültülü veriyi (sinyali) temizlemek için algoritmik bir yöntem önerilmiştir.

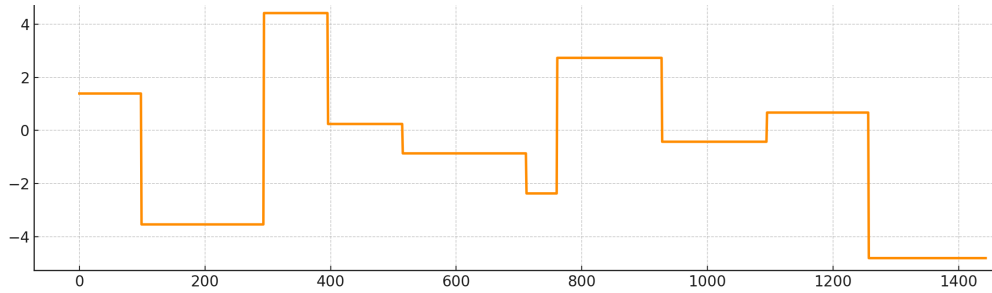


x gürültülü veriyi içeren dizi, y gürültü temizlenmiş veriyi içeren diziyi oluşturmak üzere:

- I. Her bir i indeksi için, $y_i = \text{ortanca}(x_{i-k}, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{i+k})$ ($k = 5$)
- II. y dizisinde, ardışık ve birbirine yakın değerler belirli bir eşik değeri altında gruplanır
- III. Her grubun gürültülü sinyaldeki değerlerinin ortalaması alınarak, grubun tüm elemanları bu ortalama ile değiştirilir.

Not: Bir dizideki elemanlar sıralandığında ortadaki değer ortanca (medyan) değer olur.

Gürültü temizleme yöntemi çalıştıktan sonra sonuç bu şekilde oluşmalıdır.



Verilen bu gürültü temizleme yöntemi için aşağıdakilerden hangisi söylenemez?

- A) I. adımda x_{i-k} veya x_{i+k} dizi dışında kaldığı durumda, dizinin başındaki ve sonundaki değerler kullanılır böylece dizi dışında bir indekse erişim yapılma hatası oluşmaz.
- B) $k = 5$ için ortanca değer 11 eleman üzerinden hesaplanır.
- C) II. adımda eşik değeri y dizinde bir gruptan diğer gruba geçişteki farktan düşük olacak şekilde seçilir.
- D) II. adımda ardışık ve benzer ortanca değerleri seçilen eşik altında ise gruplanarak, sinyalin doğal grup sınırları belirlenmiş olur.
- E) III. adım, her bir grup için orijinal sinyalden ortalama almayı gerektirdiğinden, bu önerilen yöntem içerisindeki en yavaş adımdır.

A

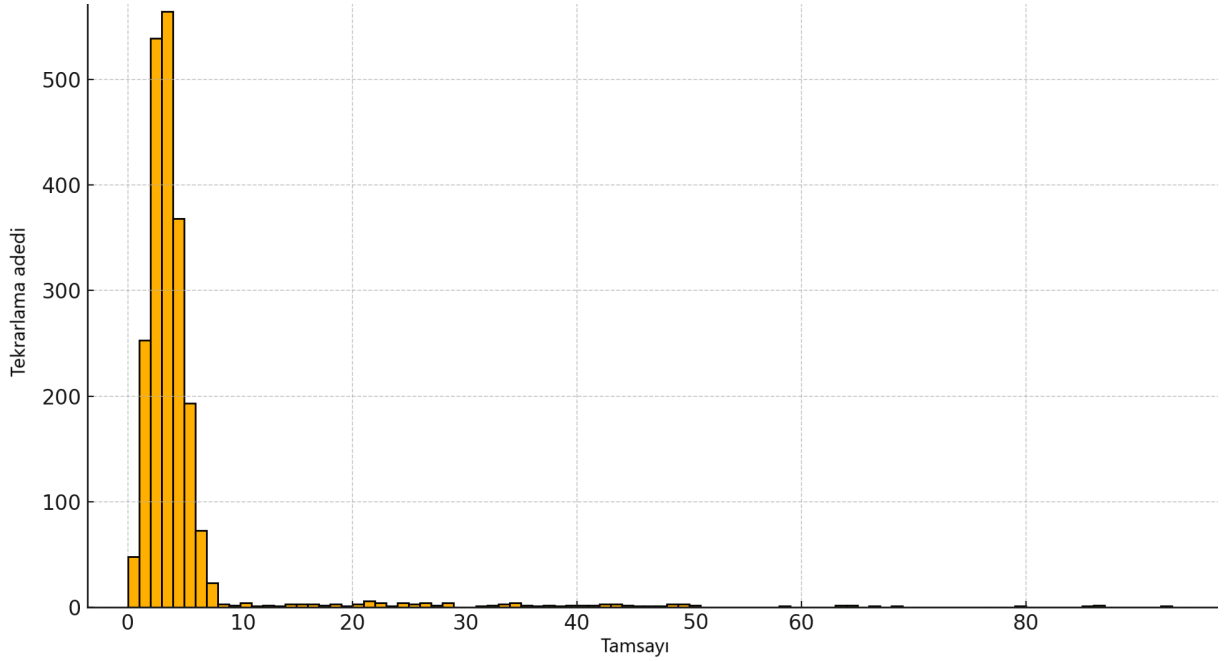
ÇÖZÜM

III. adımda gruplar için ortalama almak gerekir, ancak bu işlem zaman açısından basittir (doğrusal karmaşıklık). Yöntemin en yavaş kısmı, I. adım olan medyan filtrelemedir, çünkü pencere içerisindeki ortanca değer (sıralama, quick-select) bulma işlemi algoritmadaki en yavaş adımdır. Doğru cevap E şıkkıdır.

CEVAP: E

SORU 29

Bir cihazdan milyonlarca negatif olmayan tamsayı değeri alınmakta ve *sayilar* isminde bir dizide saklanmaktadır. Aşağıda grafikte bu sayıların binlercesi için tekrarlanma adetleri gösterilmektedir.



sayilar dizisinin etkin sıralanması için aşağıdaki şıklardan hangisi verimli hale getirilmelidir?

- A) 0 ile 10 arasındaki sayıların tekrarlanma adetlerini tutacak 11 elemanlı bir dizi oluşturulur (*histogram* isimli) ve bu dizinin her bir elemanı sıfırlanır. Ayrıca ilk değeri sıfır olan *kucuk_sayi_adet* isimli tam sayı değişken tanımlanır.
- B) Bir döngü ile dizideki elemanların hepsinin üzerinden geçilirken 0 ile 10 arasındaki değerlerin tekrarlanma adetleri bulunur (*histogram* dizisi üzerinde) ve toplam adetleri *kucuk_sayi_adet* değişkeninde toplanır. Aynı zamanda 10'dan büyük sayılar dizinin sonuna doğru atılır (yer değiştirilerek).
- C) B maddesindeki döngü bittiğinde *sayilar* dizisinin ilk *kucuk_sayi_adet* elemanı 0 ile 10 arasında olur. Bunu takip eden sayılar ise 10'dan büyük sayılar olur.
- D) Sırayla, *histogram* dizisi içindeki her bir adet kadar *sayi* dizi içerisine yazılarak 0 ile 10 arasındaki sayıların *sayilar* dizisi içine sıralı şekilde yerleştirilmesi sağlanır.
- E) En son hızlı bir sıralama algoritması tüm dizi için çalıştırılır. Dizinin ilk *kucuk_sayi_adet* kadar elemanı zaten sıralı olduğu dizinin büyük kısmını oluşturduğu için sıralama işlemi verimli bir şekilde tamamlanmış olur.

A

ÇÖZÜM

Dizinin büyük kısmını oluşturan ilk *kucuk_sayi_adet* kadarlık kısmı zaten D maddesi ile sıralı halde oluşturulduğu için E şıkkında söylendiği üzere tüm dizinin sıralanması yerine dizinin *kucuk_sayi_adet* kadar olan kısmından sonrasında yer alan son kısmının sıralanması yeterlidir. Doğru cevap E şıkkıdır.

B kitapçığındaki basım hatasından dolayı E şıkkı B kitapçığında çıkmamıştır. Bu nedenle bu sorunun İPTAL edilmesine karar verilmiştir.

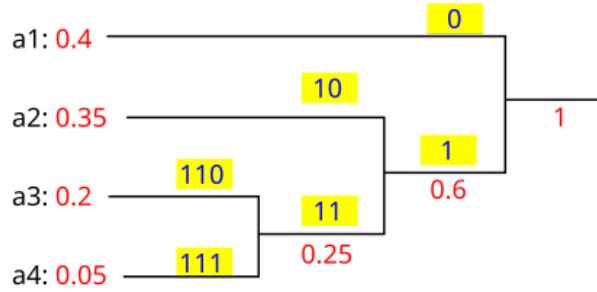
CEVAP: İPTAL

A

SORU 30

Huffman kodlama, veri sıkıştırma amacıyla kullanılan kayıpsız bir kodlama yöntemidir ve yöntemin temel adımları şunlardır:

- Kodlanacak her sembolün olasılığı hesaplanır. Sembolün olasılığı veride o sembolün kaç defa tekrar ettiği değerinin toplam veri büyüklüğüne bölünmesi olarak düşünülebilir.
- Tüm semboller frekanslarına göre sıralı olacak şekilde bir veri yapısı içine yerleştirilir.
- Ağaç Oluştur:
 - o En küçük iki frekanslı düğüm kuyruktan çıkarılır.
 - o Bu iki düğümü çocuk olarak alan, toplam frekanslı yeni bir düğüm oluşturulur.
 - o Yeni düğüm iki çocuğun toplam olasılığı ile tekrar kuyruğa eklenir.
 - o Bu işlem tek bir düğüm kalana kadar tekrar edilir (en son toplam olasılık 1 olur).



- Kod Atama:
 - o Oluşan ikili ağaç üzerinden, her yaprak düğüme (sembole) bir bit dizisi atanır.
 - o Bu atama sırasında ağaçta sol (üst) dal için 0, sağ (alt) dal için 1 eklenir.
- Kodlama:
 - o Orijinal veriyi, her sembol yerine karşılık gelen Huffman koduyla değiştirerek sıkıştırılmış veri elde edilir.

Aşağıdakilerden hangisi Huffman kodlama algoritmasının temel adımlarından biri değildir?

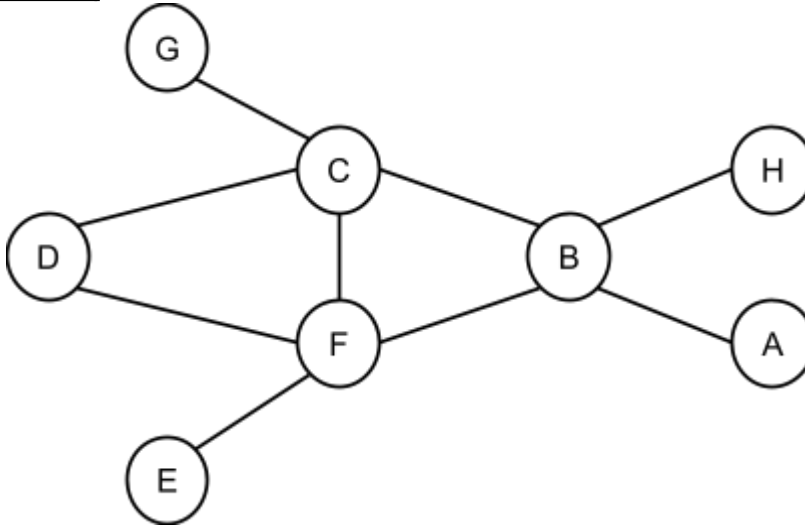
- A) Sembollerin frekanslarını belirlemek.
- B) Frekanslara göre ikili bir ağaç oluşturmak.
- C) Kod uzunluğunu azaltmak için ağacı derinlik öncelikli gezmek.
- D) En küçük frekanslı iki düğümü birleştirerek yeni düğüm oluşturmak.
- E) Ağaç üzerinde gezinerek her sembole bit dizisi atamak.

ÇÖZÜM

Huffman algoritmasında ağaç oluşturulduktan sonra kodların ağaç üzerinden hangi sıralamayla gezinerek alındığı önemli değildir. Buradaki önemli olan, kodların ağacın yapısına göre atanmasıdır. Huffman kodlamada kod uzunluğu zaten inşa edilen ağaç yapısı sayesinde azaltılır. Doğru cevap C şıkkıdır.

CEVAP: C

SORU 31



Yukarıdaki çizgede, C düğümünden başlayarak Genişlik Öncelikli Arama (Breadth-First Search - BFS) yapıldığında, aşağıdaki düğüm gezilerinden kaç tanesi görülebilir?

- C → B → D → F → G → A → H → E
- C → D → B → F → G → H → A → E
- C → G → D → F → B → E → A → H
- C → B → F → A → D → G → H → E
- C → B → H → A → F → E → D → G
- C → D → F → B → H → A → E → G

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

ÇÖZÜM

BFS başlangıç düğümünden itibaren düğümleri seviye-seviye gezer. Yani bu algoritmaya göre ilk olarak başlangıç düğüm (seviye 1), sonra çocuk düğümler (seviye 2), sonra çocuk düğümlerin ilk ziyaret sırasına göre çocukları (seviye 3) vs. şeklinde düğümler gezilir. Buna göre ilk 3 sırada verilen düğümlerin gezme sırası doğrudur.

CEVAP: C

SORU 32

- A: [B]
 B: [A, C]
 C: [E]
 D: [A, C]
 E: [G]
 F: [C]
 G: [F]

Yukarıda komşuluk listesi ile temsil edilen yönlü bir çizgede, aşağıdaki düğüm çiftlerinden kaç tanesi aynı güçlü bağlı bileşen (Strongly Connected Component) içinde yer almaz? Not: Komşuluk listesi ile temsilde bir X düğümünden Y düğümüne giden bir kenar var ise bu X: [Y] şeklinde gösterilir.

Düğüm Çiftleri:

- A ve B
- C ve E
- D ve F
- E ve G
- A ve C

- A) 1
 B) 2
 C) 3
 D) 4
 E) 5

ÇÖZÜM

A $\rightarrow$ B ve B $\rightarrow$ A olduğu için A ve B arasında karşılıklı bir bağlantı vardır. Bu nedenle A ve B aynı SCC içindedir.

C $\rightarrow$ E $\rightarrow$ G $\rightarrow$ F $\rightarrow$ C şeklinde bir döngü bulunmaktadır. Bu, C, E, G ve F düğümlerinin birbirlerine ulaşılabilir olduğunu gösterir. Dolayısıyla C, E, G ve F aynı SCC içindedir.

D $\rightarrow$ A ve D $\rightarrow$ C bağlantıları vardır. Ancak çizgedeki herhangi bir düğümden D'ye doğrudan bir dönüş yolu garanti edilmez. Bu nedenle D tek başına bir SCC oluşturur. Dolayısıyla D ve F farklı SCC içindedir.

E $\rightarrow$ G ve G $\rightarrow$ F ve F $\rightarrow$ C $\rightarrow$ E olduğu için E ve G aynı SCC içindedir.

A $\rightarrow$ B $\rightarrow$ C ile A'dan C'ye gidilebilir ancak C'den A'ya herhangi bir yol yoktur. Bu nedenle A ve C farklı SCC içindedir.

Sonuç olarak toplam 2 düğüm çifti farklı SCC içinde yer almaktadır.

CEVAP: B

SORU 33

- 0: [1, 2]
1: [3]
2: [3]
3: [4, 5]
4: []
5: []

Yukarıda komşuluk listesi ile temsil edilen yönlü bir çizgenin kaç tane farklı topolojik sıralaması vardır? Not: Komşuluk listesi ile temsilde bir X düğümünden Y düğümüne giden bir kenar var ise bu X: [Y] şeklinde gösterilir.

- A) 1
B) 2
C) 3
D) 4
E) 5

ÇÖZÜM

Bu çizgede toplam 4 farklı topolojik sıralama vardır ve bu sıralamalar şu şekildedir:

- 0 -> 1 -> 2 -> 3 -> 4 -> 5
0 -> 1 -> 2 -> 3 -> 5 -> 4
0 -> 2 -> 1 -> 3 -> 4 -> 5
0 -> 2 -> 1 -> 3 -> 5 -> 4

CEVAP: D

SORU 34

Derinlik öncelikli arama (Depth First Search - DFS) ve Genişlik öncelikli arama (Breadth First Search - BFS) ile ilgili aşağıdaki ifadelerden kaç tanesi doğrudur?

- BFS ve DFS'in teorik çalışma zamanları eşittir.
- DFS, bir düğümün tüm komşularını ziyaret ettikten sonra bir sonraki seviyeye geçer.
- DFS, çizgelerde döngü tespiti için kullanılabilir.
- DFS, yönsüz ve ağırlıksız çizgelerde başlangıç düğümünden diğer tüm düğümlere kenar sayısı açısından en kısa yolu bulmayı garanti eder.

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

ÇÖZÜM

Bu ifadeleri inceleyelim.

- BFS ve DFS'in teorik çalışma zamanları eşittir: **Doğru**. Her iki algoritmanın çalışma zamanı $\Theta(V+E)$ 'dir.
- DFS, bir düğümün tüm komşularını ziyaret ettikten sonra bir sonraki seviyeye geçer: **Yanlış**. Bu ifade BFS için doğrudur.
- DFS, çizgelerde döngü tespiti için kullanılabilir: **Doğru**. Ters kenar (Back edge) tespit edildiğinde çizgede döngü vardır.
- DFS, yönsüz ve ağırlıksız çizgelerde başlangıç düğümünden diğer tüm düğümlere kenar sayısı açısından en kısa yolu bulmayı garanti eder: **Yanlış**. Bu ifade BFS için doğrudur.

CEVAP: C

SORU 35

Verilen yönsüz, ağırlıksız ve basit bir çizge, A komşuluk matrisi ile temsil edilsin. Buna göre aşağıdaki ifadelerin kaç tanesi doğrudur?

- Eğer $A[i][j]=1$ ise, her zaman $A[j][i]=1$ olur.
- A matrisi her zaman simetriktir.
- Bir düğümün derecesi (degree) A matrisinin o satırındaki 1'lerin sayısı ile bulunabilir.
- Çizgedeki düğüm sayısı arttıkça, A'nın bellekte kapladığı yerin verimliliği komşuluk listesine göre daha fazla artar.

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

ÇÖZÜM

Bu ifadeleri inceleyelim.

- Eğer $A[i][j]=1$ ise, her zaman $A[j][i]=1$ olur: **Doğru**. Çizge yönsüz olduğu için bu ifade doğrudur.
- A matrisi her zaman simetriktir: **Doğru**. Çizge yönsüz olduğundan dolayı komşuluk matrisi her zaman simetriktir.
- Bir düğümün derecesi (degree) A matrisinin o satırındaki 1'lerin sayısı ile bulunabilir: **Doğru**. Bir satırın içindeki 1'lerin sayısı ilgili satırdaki düğümün derecesini verir.
- Çizgedeki düğüm sayısı arttıkça, A'nın bellekte kapladığı yerin verimliliği komşuluk listesine göre daha artar: **Yanlış**. Komşuluk matrisi $\Theta(n^2)$ yer kaplar. Çok büyük ama seyrek (sparse) çizgelerde komşuluk listesi çok daha verimlidir.

CEVAP: D

A

[36-50] Sorular İçin Açıklama

- Soruları C programlama dili çerçevesinde cevaplayınız.
- Derleyici olarak gcc kullanıldığını varsayınız.
- Gerekli tüm başlık (header) dosyalarının verilen programa dahil edildiğini varsayınız.

SORU 36

```
int main() {  
    char word[5] = "ABCDE";  
    int i;  
    for(i=1; i<6;i++){  
        printf("%c", word[i-1] + i % 3);  
    }  
    return 0;  
}
```

Yukarıdaki kod parçacığı çalıştırıldığında konsola aşağıdakilerden hangisini yazdırır?

- A) BDCEG
- B) ACEGI
- C) BDFHJ
- D) ACEDF
- E) BCCFG

ÇÖZÜM

$i = 1$ değeri için `word[0]` yani "A" harfi alınarak $i \% 3$ işleminin sonucu ile toplanacak $1 \% 3 = 1$.

$A + 1 = B$ olacaktır. Döngünün ilk aşamasında ekrana **B** yazdırılacaktır.

$i = 2$ değeri için `word[1]` yani "B" harfi alınarak $i \% 3$ işleminin sonucu ile toplanacak $2 \% 3 = 2$.

$B + 2 = D$ olacaktır. Ekrana **D** yazdırılacaktır.

$i = 3$ değeri için `word[2]` yani "C" harfi alınarak $i \% 3$ işleminin sonucu ile toplanacak $3 \% 3 = 0$.

$C + 0 = C$ olacaktır. Ekrana **C** yazdırılacaktır.

$i = 4$ değeri için `word[3]` yani "D" harfi alınarak $i \% 3$ işleminin sonucu ile toplanacak $4 \% 3 = 1$.

$D + 1 = E$ olacaktır. Ekrana **E** yazdırılacaktır.

$i = 5$ değeri için `word[4]` yani "E" harfi alınarak $i \% 3$ işleminin sonucu ile toplanacak $5 \% 3 = 2$.

$E + 2 = G$ olacaktır. Ekrana **G** yazdırılacaktır.

SORU 37

```
int main() {  
    int i, count = 0;  
    for(i=1; i<17; i=i+i){  
        int j = 2;  
        do {  
            count++;  
            j*=2;  
        } while (j<=i);  
    }  
    printf("%d\n", count);  
    return 0;  
}
```

Yukarıdaki kod parçacığı çalıştırıldığında programın çıktısı aşağıdakilerden hangisi olur?

- A) 8
- B) 9
- C) 10
- D) 11
- E) 12

ÇÖZÜM

Program i: 1, j:2 değerleri ile başlar ve içteki döngü ile devam eder. *count* 1 artar ve içteki döngü sonlanır. i değeri 2 olur ve j değeri de 2'den başlar.

i: 2, j:2 için *count* 1 artar ve j 4 olur içteki döngü sonlanır. i değeri 4 olur ve j değeri 2'den başlar.

i: 4, j:2 için *count* 1 artar ve j 4 olur ve içteki döngü devam eder.

i: 4, j:4 için *count* 1 artar ve j 8 olur ve içteki döngü sona erer. i değeri 8 olur ve j değeri 2'den başlar.

i: 8, j:2 için *count* 1 artar ve j 4 olur ve içteki döngü devam eder.

i: 8, j:4 için *count* 1 artar ve j 8 olur ve içteki döngü devam eder.

i: 8, j:8 için *count* 1 artar ve j 16 olur ve içteki döngü sona erer. i değeri 16 olur ve j değeri 2'den başlar.

i: 16, j:2 için *count* 1 artar ve j 4 olur ve içteki döngü devam eder.

A

i: 16, j:4 için count 1 artar ve j 8 olur ve içteki döngü devam eder.

i: 16, j:8 için count 1 artar ve j 16 olur ve içteki döngü devam eder.

i: 16, j:16 için count 1 artar ve j 32 olur ve içteki döngü sona erer. Daha sonrasında i 32 olur ve dıştaki döngü de sona erer. Program sonunda count değeri 11 olur.

CEVAP: D

SORU 38

```
int main() {
    int A[9] = {5,3,6,2,4,1,8,7,9};
    int i;
    for(i = 0; i < 7; i++){
        if(i % 2 == 0){
            if(A[i] < A[i+2]){
                int j = A[i];
                A[i] += A[i+2];
                A[i] -= j;
                A[i+2] = j;
            }
            else{
                int j = A[i+1];
                A[i+1] += A[i];
                A[i+1] -= j;
                A[i] = j;
            }
        }
        else{
            int j = A[i-1];
            A[i-1] += A[i];
            A[i-1] -= j;
            A[i] = j;
        }
    }
    for(i = 0; i < 9 ; i++){
        printf("%d ", A[i]);
    }
    return 0;
}
```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) 3 6 5 2 1 8 4 7 9
- B) 3 6 2 5 4 1 8 7 9
- C) 3 6 5 2 8 1 4 7 9
- D) 6 3 5 2 4 1 8 7 9
- E) 3 6 5 2 1 8 9 7 4

A

ÇÖZÜM

Döngünün başında i değeri 0 ile başlar ve $0\%2 = 0$ doğru olduğundan if bloğu çalışır.

Daha sonrasında $A[0] < A[2]$ olduğundan içerideki if çalışır ve if blokları içindeki kod satırları sadece eleman değişikliği yapmaktadır. Yani 0 ve 2. elemanlar yer değiştirilir.

6 3 5 2 4 1 8 7 9

Sonraki iterasyonda $i = 1$ olur ve else kısmı çalışır. Burada $A[0]$ ve $A[1]$ yer değiştirilir.

3 6 5 2 4 1 8 7 9

Sonraki iterasyonda $i = 2$ olur ve if bloğu içerisindeki else kısmı çalışır $A[2] < A[4]$ ifadesi yanlıştır. Burada $A[2]$ ve $A[3]$ elemanlarının yerleri değişir.

3 6 2 5 4 1 8 7 9

Sonraki iterasyonda $i = 3$ olur ve else kısmı çalışır. Burada $A[3]$ ve $A[2]$ yer değiştirilir.

3 6 5 2 4 1 8 7 9

Sonraki iterasyonda $i = 4$ olur ve if bloğu içerisindeki if kısmı çalışır $A[4] < A[6]$ ifadesi doğrudur. Burada $A[4]$ ve $A[6]$ elemanlarının yerleri değişir.

3 6 5 2 8 1 4 7 9

Sonraki iterasyonda $i = 5$ olur ve else kısmı çalışır. Burada $A[5]$ ve $A[4]$ yer değiştirilir.

3 6 5 2 1 8 4 7 9

Son iterasyonda $i = 6$ olur ve if bloğu içerisindeki if kısmı çalışır $A[6] < A[8]$ ifadesi doğrudur. Burada $A[6]$ ve $A[8]$ elemanlarının yerleri değişir.

3 6 5 2 1 8 9 7 4

CEVAP: E

SORU 39

```

#define MAX_SIZE 9
void insert(int arr[MAX_SIZE], int value) {
    int index = 0;
    while (index != -1) {
        if (arr[index] == 0) {
            arr[index] = value;
            return;
        }
        if (value < arr[index]) {
            int ind = 2 * index + 1;
            if (ind >= MAX_SIZE) return;
            index = ind;
        }
        else {
            int ind = 2 * index + 2;
            if (ind >= MAX_SIZE) return;
            index = ind;
        }
    }
}
int main() {
    int arr[9] = {0,0,0,0,0,0,0,0,0};
    insert(arr,6);
    insert(arr,8);
    insert(arr,9);
    insert(arr,4);
    insert(arr,5);
    insert(arr,2);
    insert(arr,7);
    insert(arr,3);
    insert(arr,1);
    int i;
    for(i=0; i<9; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}

```

Yukarıdaki kod parçası çalıştığında ekrana aşağıdakilerden hangisini yazdırır?

- A) 6 4 8 2 5 7 9 1 3
- B) 6 4 8 2 5 1 3 7 9
- C) 6 4 8 9 1 2 3 7 5
- D) 6 4 8 9 2 5 1 3 7
- E) 6 4 8 1 3 2 5 7 9

A

ÇÖZÜM

Verilen insert fonksiyonu binary search tree (ikili arama ağacı)'nı diziler ile oluşturacak şekilde tasarlanmıştır.

insert(arr,6) sonucunda 6 ilk eleman olarak diziye eklenir.

6 0 0 0 0 0 0 0

insert(arr,8) sonucunda 8 arr[2] indeksine eklenir.

6 0 8 0 0 0 0 0

insert(arr,9) sonucunda 9 arr[6] indeksine eklenir.

6 0 8 0 0 0 9 0

insert(arr,4) sonucunda 4 arr[1] indeksine eklenir.

6 4 8 0 0 0 9 0

insert(arr,5) sonucunda 5 arr[4] indeksine eklenir.

6 4 8 0 5 0 9 0

insert(arr,2) sonucunda 2 arr[3] indeksine eklenir.

6 4 8 2 5 0 9 0

insert(arr,7) sonucunda 7 arr[5] indeksine eklenir.

6 4 8 2 5 7 9 0

insert(arr,3) sonucunda 3 arr[8] indeksine eklenir.

6 4 8 2 5 7 9 0 3

insert(arr,1) sonucunda 1 arr[7] indeksine eklenir.

6 4 8 2 5 7 9 1 3

CEVAP: A

SORU 40

```
int f1(int n) {
    int c = 0;
    while (n) {
        c += n & 1;
        n >>= 1;
    }
    return c;
}
int f2(int n, int m) {
    return n ^ (1 << m);
}
int main() {
    int n = f2(26, 2);
    printf("%d", f1(n));
    return 0;
}
```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 7

A

ÇÖZÜM

26'nın ikiliye dönüşümünü şu şekilde adım adım yapabiliriz:

$$26 \div 2 = 13, \text{ kalan } 0$$

$$13 \div 2 = 6, \text{ kalan } 1$$

$$6 \div 2 = 3, \text{ kalan } 0$$

$$3 \div 2 = 1, \text{ kalan } 1$$

$$1 \div 2 = 0, \text{ kalan } 1$$

Son olarak, kalanları alttan üste doğru sıralarsak 11010 ikili sayısı elde edilir.

Yani, 26 sayısının ikili (binary) karşılığı **11010**'dur.

f2(26,2) şeklinde çağırıldığında

1 << 2 şeklinde shift edilip 100 olur. 26 (11010) ile 100 xor operasyonu yapıldığında:

$$\begin{array}{r} 11010 \\ 00100 \\ \text{xor}----- \\ 11110 \end{array}$$

f1 fonksiyonu bu değerle çağırıldığında döngünün her aşamasında son basamak 1 ile and işlemine tabi tutulup sonuç bulunur ve bir kez sağ shift işlemi ile son basamaktan kurtulur.

$$\begin{array}{r} 11110 \\ 00001 \\ \&----- \\ 0 \end{array} + \begin{array}{r} 1111 \\ 0001 \\ \&----- \\ 1 \end{array} + \begin{array}{r} 111 \\ 001 \\ \&----- \\ 1 \end{array} + \begin{array}{r} 11 \\ 01 \\ \&----- \\ 1 \end{array} + \begin{array}{r} 1 \\ 1 \\ \&----- \\ 1 \end{array} = 4$$

CEVAP: C

SORU 41

```
#define SIZE 4
void func(int matrix[SIZE][SIZE], int result[SIZE][SIZE])
{
    int i, j;
    for (i = 1; i < SIZE - 1; i++) {
        for (j = 1; j < SIZE - 1; j++) {
            int t = 0;
            t += matrix[i-1][j];
            t += matrix[i+1][j];
            t += matrix[i][j-1];
            t += matrix[i][j+1];
            result[i][j] = t;
        }
    }
}

int main() {
    int matrix[SIZE][SIZE] = {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12},
        {13, 14, 15, 16}
    };
    int sonuc[SIZE][SIZE] = {0};
    func(matrix, sonuc);
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            printf("%d\n", sonuc[i][j]);
        }
    }
    return 0;
}
```

Yukarıdaki kod parçası çalıştığında her satırda ekrana sonuc dizisinin bir elemanını yazdırmaktadır. Program aşağıdaki değerlerden hangisini ekrana yazdırmaz?

- A) 24
- B) 28
- C) 36
- D) 40
- E) 44

A

ÇÖZÜM

Verilen fonksiyona matris gönderildiğinde iç içe olan iki döngü ilk olarak

$i = 1, j = 1$ değeri için çalışacak ve

$\text{matrix}[i-1][j]$ yani $\text{matrix}[0][1] = 2$

$\text{matrix}[i+1][j]$ yani $\text{matrix}[2][1] = 10$

$\text{matrix}[i][j-1]$ yani $\text{matrix}[1][0] = 5$

$\text{matrix}[i][j+1]$ yani $\text{matrix}[1][2] = 7$ değerlerinin toplamı $2+10+5+7 = 24$

$i = 1, j = 2$ değeri için çalışacak ve

$\text{matrix}[i-1][j]$ yani $\text{matrix}[0][2] = 3$

$\text{matrix}[i+1][j]$ yani $\text{matrix}[2][2] = 11$

$\text{matrix}[i][j-1]$ yani $\text{matrix}[1][1] = 6$

$\text{matrix}[i][j+1]$ yani $\text{matrix}[1][3] = 8$ değerlerinin toplamı $3+11+6+8 = 28$

$i = 2, j = 1$ değeri için çalışacak ve

$\text{matrix}[i-1][j]$ yani $\text{matrix}[1][1] = 6$

$\text{matrix}[i+1][j]$ yani $\text{matrix}[3][1] = 14$

$\text{matrix}[i][j-1]$ yani $\text{matrix}[2][0] = 9$

$\text{matrix}[i][j+1]$ yani $\text{matrix}[2][2] = 11$ değerlerinin toplamı $6+14+9+11 = 40$

$i = 2, j = 2$ değeri için çalışacak ve

$\text{matrix}[i-1][j]$ yani $\text{matrix}[1][2] = 7$

$\text{matrix}[i+1][j]$ yani $\text{matrix}[3][2] = 15$

$\text{matrix}[i][j-1]$ yani $\text{matrix}[2][1] = 10$

$\text{matrix}[i][j+1]$ yani $\text{matrix}[2][3] = 12$ değerlerinin toplamı $7+15+10+12 = 44$

olacak ve döngüler sonlanacaktır. Matrisin diğer elemanları da 0 olarak kalacaktır.

Yani program 36 değerini ekrana yazdıramayacaktır.

CEVAP: C

SORU 42

```
int main() {  
    int i,j,k,count=0;  
  
    for(i=1; i<100; i=i<<1){  
        for(j = i++; j<100; j=j<<2)  
            count++;  
    }  
    printf("%d", count);  
    return 0;  
}
```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) 10
- B) 11
- C) 12
- D) 13
- E) 14

ÇÖZÜM

Döngülerin içerisindeki count++ işlemi sadece aşağıdaki i ve j değerleri için toplamda **13** kez çalışacaktır.

i:2, j:1
i:2, j:4
i:2, j:16
i:2, j:64
i:5, j:4
i:5, j:16
i:5, j:64
i:11, j:10
i:11, j:40
i:23, j:22
i:23, j:88
i:47, j:46
i:95, j:94

CEVAP: D

A

SORU 43

```
int f1(int *arr, int size) {
    if (size == 1) {
        return *arr;
    }
    int m = f1(arr + 1, size - 1);
    return (*arr > m) ? *arr : m;
}
int f2(int *arr, int size, int t) {
    if (size == 0) return 0;
    return (*arr == t) + f2(arr + 1, size - 1, t);
}
int main() {
    int arr[] = {1, 9, 2, 3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3};
    int size = 20;
    printf("%d", f2(arr, size, f1(arr, size)));
    return 0;
}
```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

A

ÇÖZÜM

f1 fonksiyonu bir dizi içerisindeki en büyük sayıyı bulmaya yaramaktadır. f1(arr,20) şeklinde çağırıldığında ilk eleman diziden ayrılıp kalan kısmın maksimum değeri yine aynı fonksiyon ile hesaplanır ve hangisi daha büyükse fonksiyon o değeri return eder. Sonuçlar aşağıdan yukarı doğru gelir.

- 1 ile kalan dizinin maksimumu {9, 2, 3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 9 ile kalan dizinin maksimumu {2, 3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 2 ile kalan dizinin maksimumu {3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 3 ile kalan dizinin maksimumu {9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 9 ile kalan dizinin maksimumu {2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 2 ile kalan dizinin maksimumu {8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 8 ile kalan dizinin maksimumu {4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 4 ile kalan dizinin maksimumu {9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 9 ile kalan dizinin maksimumu {6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 9
- 6 ile kalan dizinin maksimumu {3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 7
- 3 ile kalan dizinin maksimumu {2, 6, 2, 3, 5, 6, 7, 2, 3} □ 7
- 2 ile kalan dizinin maksimumu {6, 2, 3, 5, 6, 7, 2, 3} □ 7
- 6 ile kalan dizinin maksimumu {2, 3, 5, 6, 7, 2, 3} □ 7
- 2 ile kalan dizinin maksimumu {3, 5, 6, 7, 2, 3} □ 7
- 3 ile kalan dizinin maksimumu {5, 6, 7, 2, 3} □ 7
- 5 ile kalan dizinin maksimumu {6, 7, 2, 3} □ 7
- 6 ile kalan dizinin maksimumu {7, 2, 3} □ 7
- 7 ile kalan dizinin maksimumu {2, 3} □ 7
- 2 ile kalan dizinin maksimumu {3} □ 3

Daha sonra f2 fonksiyonu da f2(arr, 20, 9) şeklinde çağırılır. Bu fonksiyonun amacı da verilen sayının dizi içerisinde kaç kez geçtiğini hesaplamaktır. Sonuçlar aşağıdan yukarı doğru hesaplanarak gelir.

- 1 ile t (9) eşit mi {9, 2, 3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 3
- 9 ile t (9) eşit mi {2, 3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 3
- 2 ile t (9) eşit mi {3, 9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 2
- 3 ile t (9) eşit mi {9, 2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 2
- 9 ile t (9) eşit mi {2, 8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 2
- 2 ile t (9) eşit mi {8, 4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 1
- 8 ile t (9) eşit mi {4, 9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 1
- 4 ile t (9) eşit mi {9, 6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 1
- 9 ile t (9) eşit mi {6, 3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 1
- 6 ile t (9) eşit mi {3, 2, 6, 2, 3, 5, 6, 7, 2, 3} □ 0
- 3 ile t (9) eşit mi {2, 6, 2, 3, 5, 6, 7, 2, 3} □ 0
- 2 ile t (9) eşit mi {6, 2, 3, 5, 6, 7, 2, 3} □ 0
- 6 ile t (9) eşit mi {2, 3, 5, 6, 7, 2, 3} □ 0
- 2 ile t (9) eşit mi {3, 5, 6, 7, 2, 3} □ 0
- 3 ile t (9) eşit mi {5, 6, 7, 2, 3} □ 0
- 5 ile t (9) eşit mi {6, 7, 2, 3} □ 0
- 6 ile t (9) eşit mi {7, 2, 3} □ 0
- 7 ile t (9) eşit mi {2, 3} □ 0
- 2 ile t (9) eşit mi {3} □ 0
- { } size 0 olduğu için sonuç □ 0

CEVAP: C

SORU 44

```

#define SIZE 5
int main() {
    char matrix[SIZE][SIZE] = {0};

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            if (j == i && i <= SIZE / 2) {
                matrix[i][j] = 1;
            }
            if (j == SIZE - i - 2 && i <= SIZE / 2) {
                matrix[i][j] = 1;
            }
            if (i > SIZE / 2 && j == SIZE / 2) {
                matrix[i][j] = 1;
            }
        }
    }
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

Yukarıdaki kod parçacığı çalıştığında aşağıdakilerden hangisi ekrana yazdırılır?

A) 1 0 0 1 0
0 1 1 0 0
0 1 1 0 0
0 0 1 0 0
0 0 1 0 0

B) 1 0 0 1 0
0 1 1 0 0
0 1 1 0 0
1 0 1 1 0
0 0 1 0 1

C) 1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0

D) 1 0 1 0 0
0 1 0 0 0
1 0 1 0 0
0 0 1 0 0
0 0 1 0 0

E) 1 0 1 0 1
0 1 0 1 0
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0

A

ÇÖZÜM

Döngülerin $i=0$ ve $j=0$ değeri ile başlayacak ve aşağıdaki gibi ilerleyecektir.

$i:0, j:0$

$i:0, j:1$

$i:0, j:2$

$i:0, j:3$

$i:0, j:4$

$i:1, j:0$

$i:1, j:1$

$i:1, j:2$

$i:1, j:3$

$i:1, j:4$

$i:2, j:0$

$i:2, j:1$

$i:2, j:2$

$i:2, j:3$

$i:2, j:4$

$i:3, j:0$

$i:3, j:1$

$i:3, j:2$

$i:3, j:3$

$i:3, j:4$

$i:4, j:0$

$i:4, j:1$

$i:4, j:2$

$i:4, j:3$

$i:4, j:4$

İlk if bloğu i 'nin j 'ye eşit olduğu ve i 'nin $SIZE/2$ ye küçük eşit olduğu yani 2 ye küçük eşit olduğu durumlarda dizinin elemanını 1 yapar. Sadece aşağıdaki durumlar:

$i:0, j:0$

$i:1, j:1$

$i:2, j:2$

1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

0 0 0 0 0

0 0 0 0 0

İkinci if bloğu j değerini $SIZE - i - 2$ olarak başlatır ve $i \leq SIZE / 2$ oldukça devam eder.

$j = 3 - i$ olarak başlar ve $i \leq 2$ iken çalışır. Bu durumda:

$i = 0$ için $j = 3$

$i = 1$ için $j = 2$

$i = 2$ için $j = 1$ durumlarında dizinin elemanları 1 yapılır.

1 0 0 1 0

0 1 1 0 0

0 1 1 0 0

0 0 0 0 0

0 0 0 0 0

A

Son if bloğu $i > \text{SIZE} / 2$ yani $i > 2$ ve $j = \text{SIZE} / 2$ yani $j = 2$ olduğunda çalışır. Bu durumda:

$i = 3$ için $j = 2$

$i = 4$ için $j = 2$ durumlarında dizinin elemanları 1 yapılır.

1 0 0 1 0

0 1 1 0 0

0 1 1 0 0

0 0 1 0 0

0 0 1 0 0

CEVAP: A

SORU 45

```

int f2(int a, int b) {
    if (a == 0)
        return b;
    return f2(a - 1, b + a);
}
int f1(int x, int y){
    if (--x > y++ && x-- == ++y) {
        return f2(x,y);
    } else {
        return f2(y,x);
    }
}
int main() {
    int x = 5, y = 2;
    printf("%d", f1(x, y));
    return 0;
}

```

Yukarıdaki kod parçası ekrana aşağıdakilerden hangisini yazdırır?

- A) 7
- B) 8
- C) 10
- D) 11
- E) 13

ÇÖZÜM

Program f1(5,2) değeri ile çağırılır. f1 içerisindeki if kısmında:

if (--x > y++ && x-- == ++y)

--x > y++

4 > 2 doğru olur. y kullanımdan sonra artar.

x-- == ++y

4 == 4 doğru olur. x kullanımdan sonra azalır.

Doğru && Doğru işleminin sonucu Doğru olur ve if bloğu çalışır.

Program f2(3,4) sonucunu yazdırır.

f2(3, 4) sonucunu hesaplarsak.

a = 0 olduğunda b değeri dönecek ancak değilse f2(a-1,b+a) hesaplanacak.

f2(3,4) = f2(2,3+4), f2(2,7)

f2(2,7) = f2(1, 2+7) = f2(1,9)

f2(1,9) = f2(0, 1+9) = f2(0,10)

f2(0,10) = **10**

CEVAP: C

SORU 46

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr1 = arr + 3;  
    int *ptr2 = ptr1 - 2;  
  
    printf("%d\n", *(ptr2 + (*ptr1 - *ptr2)));  
  
    return 0;  
}
```

Yukarıdaki kodun ekran çıktısı nedir?

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

ÇÖZÜM

Bu kodun açıklaması şu şekildedir:

```
int arr[] = {1, 2, 3, 4, 5};
```

Bu satırda, arr adında bir tamsayı dizisi (integer array) oluşturuluyor ve başlangıç değerleri atanıyor. Dizi, [1, 2, 3, 4, 5] elemanlarını içeriyor.

```
int *ptr1 = arr + 3;
```

ptr1 adında bir tamsayı göstericisi (integer pointer) oluşturuluyor.

arr + 3 ifadesi, arr dizisinin başlangıç adresine 3 tamsayılık yer ekliyor. Bu, ptr1'in arr dizisinin 4. elemanını (diziler 0'dan başladığı için arr[3]) işaret etmesini sağlıyor. Yani, ptr1'in gösterdiği değer 4'tür.

```
int *ptr2 = ptr1 - 2;
```

ptr2 adında başka bir tamsayı göstericisi oluşturuluyor.

ptr1 - 2 ifadesi, ptr1'in gösterdiği adresten 2 tamsayılık yer çıkarıyor. Bu, ptr2'nin arr dizisinin 2. elemanını (diziler 0'dan başladığı için arr[1]) işaret etmesini sağlıyor. Yani, ptr2'nin gösterdiği değer 2'dir.

```
*ptr1
```

ptr1'in gösterdiği adresteki değeri verir. Bu, 4'tür.

```
*ptr2
```

ptr2'in gösterdiği adresteki değeri verir. Bu, 2'dir.

A

```
*ptr1 - *ptr2
```

4 - 2 işleminin sonucunu verir, yani 2'dir.

```
ptr2 + (*ptr1 - *ptr2)
```

ptr2'nin gösterdiği adrese 2 tamsayılık yer ekliyor. ptr2 başlangıçta arr[1]'i gösteriyordu, bu nedenle ptr2 + 2, arr[3]'ü gösterir.

```
*(ptr2 + (*ptr1 - *ptr2))
```

ptr2 + 2 ifadesinin gösterdiği adresteki değeri verir. Bu, arr[3]'ün değeri olan 4'tür.

```
printf("%d\n", *(ptr2 + (*ptr1 - *ptr2)));
```

Sonuç olarak, 4 ekrana yazdırılır.

CEVAP: E

SORU 47

```

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    int sum = 0;

    for (int i = 0; i < 3; i++) {
        sum += *(ptr + *(ptr + i));
    }

    printf("%d\n", sum);

    return 0;
}

```

Yukarıdaki kodun ekran çıktısı nedir?

- A) 6
- B) 7
- C) 8
- D) 9
- E) Hiçbiri

ÇÖZÜM

Kodu adım adım inceleyelim:

1. **int arr[] = {1, 2, 3, 4, 5};**: Bir tamsayı dizisi **arr** oluşturulur ve {1, 2, 3, 4, 5} değerleri ile başlatılır.
2. **int *ptr = arr;**: Bir tamsayı göstericisi **ptr** oluşturulur ve dizinin ilk elemanının adresini (yani **arr[0]**'in adresini) işaret eder. Bu durumda ***ptr** değeri 1'dir.
3. **int sum = 0;**: Bir tamsayı değişkeni **sum** oluşturulur ve başlangıç değeri 0 olarak atanır.
4. **for (int i = 0; i < 3; i++) { ... }**: Bu döngü 3 kez çalışacaktır (**i** 0, 1, ve 2 değerlerini alacaktır).
5. **sum += *(ptr + *(ptr + i));**: Döngünün her adımında **sum** değişkenine bir değer eklenecektir. Bu değeri inceleyelim:
 - ***(ptr + i)**: Bu ifade, **ptr**'in işaret ettiği adresten **i** tamsayı kadar ileri giderek o adresteki değeri alır.
 - **i = 0** için: ***(ptr + 0)** eşittir ***ptr**, yani **arr[0]**'in değeri olan 1.
 - **i = 1** için: ***(ptr + 1)** eşittir **arr[1]**'in değeri olan 2.

A

- $i = 2$ için: $*(ptr + 2)$ eşittir $arr[2]$ 'in değeri olan 3.
- $*(ptr + *(ptr + i))$: Bu ifade ise daha karmaşıktır. Önce $*(ptr + i)$ hesaplanır (yukarıda bulduk). Sonra bu değer, ptr 'in işaret ettiği adresten o değer kadar ileri gidilerek o adresteki değer alınır.
 - $i = 0$ için: $*(ptr + *(ptr + 0))$ eşittir $*(ptr + 1)$, yani $arr[1]$ 'in değeri olan 2. sum şimdi $0 + 2 = 2$ olur.
 - $i = 1$ için: $*(ptr + *(ptr + 1))$ eşittir $*(ptr + 2)$, yani $arr[2]$ 'in değeri olan 3. sum şimdi $2 + 3 = 5$ olur.
 - $i = 2$ için: $*(ptr + *(ptr + 2))$ eşittir $*(ptr + 3)$, yani $arr[3]$ 'in değeri olan 4. sum şimdi $5 + 4 = 9$ olur.
- 6. **`printf("%d\n", sum);`**: Döngü tamamlandıktan sonra sum değişkeninin değeri ekrana yazdırılır.

Bu adımlara göre, sum değişkeninin son değeri 9 olacaktır.

Dolayısıyla kodun ekran çıktısı: 9

CEVAP: D

SORU 48

```

typedef struct Nokta {
    int x;
    int y;
}N;

int main() {
    N nokta[] = {{10, 20}, {30, 40}, {50, 60}};
    N *ptr1 = nokta + 1;
    N *ptr2 = nokta + 2;
    int val = ...;

    printf("%d\n", val);

    return 0;
}

```

Yukarıdaki kod parçasında, ile gösterilen yere hangi ifade getirilmelidir ki kodun ekran çıktısı 85 olsun?

- A) ptr1->x + ptr2->y
- B) ptr1->y + ptr2->x
- C) ptr1->y + ptr2->y - 15
- D) ptr1->x + ptr2->x + 25
- E) ptr1->x + ptr1->y + ptr2->x - ptr2->y + 5

ÇÖZÜM

```
struct Point points[] = {{10, 20}, {30, 40}, {50, 60}};
```

Point struct'larından oluşan bir dizi points oluşturulur ve başlatılır.

```
struct Point *ptr1 = points + 1;
```

ptr1, points dizisinin ikinci elemanını işaret eder, yani {30, 40}.

```
struct Point *ptr2 = points + 2;
```

ptr2, points dizisinin üçüncü elemanını işaret eder, yani {50, 60}.

```
int val = ptr1->y + ptr2->y - 15;
```

ptr1->y: ptr1'in işaret ettiği struct'ın y değerini verir, yani 40.

ptr2->y: ptr2'in işaret ettiği struct'ın y değerini verir, yani 60.

ptr1->y + ptr2->y - 15: 40 + 60 - 15 = 85.

val = 85

```
printf("%d\n", val);
```

val'ın değeri ekrana yazdırılır, yani 85.

CEVAP: C

SORU 49

```
int main() {
    char str1[] = "Bilgisayar";
    char str2[] = "Olimpiyatlari";
    char str3[40];
    char *ptr = str3;

    for (int i = 0; i < 9; i += 2)
        *ptr++ = str1[i];

    for (int i = 7; i >= 0; i--)
        if (i % 2 == 0)
            *ptr++ = str2[i];

    *ptr = '\\0';

    printf("%s\\n", str3);

    return 0;
}
```

Aşağıdaki kod parçasının ekran çıktısı nedir?

- A) BliaaypiO
- B) Bliaaypiatr
- C) BliaayipO
- D) BlgaalpiO
- E) BliytaaOl pia

ÇÖZÜM

char str1[] = "Bilgisayar";
str1 dizisi "Bilgisayar" string'i ile başlatılır.

char str2[] = "Olimpiyatlari";
str2 dizisi "Olimpiyatlari" string'i ile başlatılır.

char str3[40];
str3 dizisi 40 karakterlik boş bir dizi olarak oluşturulur.

char *ptr = str3;
ptr işaretçisi str3 dizisinin başlangıç adresine atanır.

A

İlk döngü:

Bu döngü, str1 dizisinin çift indeksli karakterlerini str3 dizisine kopyalar.

i = 0: str3[0] = 'B'

i = 2: str3[1] = 'l'

i = 4: str3[2] = 'i'

i = 6: str3[3] = 'a'

i = 8: str3[4] = 'a'

str3 dizisi "Bliaa" olur.

İkinci döngü:

Bu döngü, str2 dizisinin çift indeksli karakterlerini tersten str3 dizisine kopyalar.

i = 6: str3[5] = 'y'

i = 4: str3[6] = 'p'

i = 2: str3[7] = 'i'

i = 0: str3[8] = 'O'

str3 dizisi "BliaaypiO" olur.

```
*ptr = '\0';
```

str3 dizisinin sonuna null karakteri eklenir. Bu, dizinin bir string olarak doğru şekilde sonlandırılmasını sağlar.

```
printf("%s\n", str3);
```

str3 dizisi ekrana yazdırılır.

CEVAP: A

SORU 50

```
int main() {
    int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int *ptr = &matrix[0][0];
    int sum = 0;

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if ((i + j) % 2 == 0)
                sum += *(ptr + i * 3 + j);
            else
                sum -= *(ptr + i * 3 + j);
        }
    }

    printf("%d", sum);
    return 0;
}
```

Yukarıdaki kodun ekran çıktısı ne olur?

- A) -5
- B) 0
- C) 5
- D) 10
- E) 15

A

ÇÖZÜM

matrix[3][3] dizisi şu şekilde tanımlanmıştır:

1 2 3

4 5 6

7 8 9

ptr, dizinin başlangıç adresini (matrix[0][0] = 1) gösterir.

i ve j nested loop'ları, her bir dizi elemanını gezer.

(i + j) % 2 == 0 koşulu:

Çift toplamli indekslerde (i+j çift ise) elemanı sum'a ekler.

Tek toplamli indekslerde çıkarır.

Hesaplama Tablosu:

i	j	i+j	matrix[i][j]	İşlem	sum Değişimi
0	0	0	1	sum += 1	0 + 1 = 1
0	1	1	2	sum -= 2	1 - 2 = -1
0	2	2	3	sum += 3	-1 + 3 = 2
1	0	1	4	sum -= 4	2 - 4 = -2
1	1	2	5	sum += 5	-2 + 5 = 3
1	2	3	6	sum -= 6	3 - 6 = -3
2	0	2	7	sum += 7	-3 + 7 = 4
2	1	3	8	sum -= 8	4 - 8 = -4
2	2	4	9	sum += 9	-4 + 9 = 5

Sonuç:

sum'ın final değeri 5 olur.

CEVAP: C