



KİTAPÇIK KODU : b1gl

TÜRKİYE BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA KURUMU  
BİLİM İNSANI DESTEK PROGRAMLARI BAŞKANLIĞI

34. BİLİM OLİMPİYATLARI BİRİNCİ AŞAMA SINAVI - 2026

## BİLGİSAYAR (LİSE)

Soru Kitapçığı Türü

# A

16 Mayıs 2026 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 değerde olup, dört yanlış cevap bir doğru cevabı götürcektir.** 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.
- Sınava giren adayın bir soruya itiraz etmek istemesi durumunda, sınav soruları ve cevap anahtarı TÜBİTAK'ın internet sayfasında (<https://bilimolimpiyatlari.tubitak.gov.tr>) yayımlandıktan sonra 5 iş günü içerisinde, kanıtları ile birlikte, TÜBİTAK'a başvurması gerekir. Bu tarihten sonra yapılacak başvurular işleme konmayacaktır. Sadece sınava 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 zorunda değildir, sorumluluk size aittir.
- Sınav başladıktan sonraki ilk yarım saat içinde sınav salonundan ayrılmak yasaktır.
- Sınav süresince sınava 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

## SORU 1

Bir e-posta sunucusunda Naive Bayes tabanlı bir spam filtresi çalışmaktadır. Filtre eğitim verilerinden aşağıdaki istatistikleri öğrenmiştir:

Bilgi	Değer
Gelen bir e-postanın spam olma olasılığı	%20
Spam olduğu bilinen bir e-postada "ücretsiz" kelimesinin geçme olasılığı	%60
Normal olduğu bilinen bir e-postada "ücretsiz" kelimesinin geçme olasılığı	%10
Spam olduğu bilinen bir e-postada "tıklayın" kelimesinin geçme olasılığı	%70
Normal olduğu bilinen bir e-postada "tıklayın" kelimesinin geçme olasılığı	%5

Naive Bayes'te kelimeler birbirinden koşullu bağımsız kabul edilir. İki kelime birlikte geçiyorsa olasılıklar çarpılır. Payda her sınıf için ortak olduğundan sadece pay hesaplanarak karşılaştırma yapılabilir.

Yeni gelen bir e-postada hem "ücretsiz" hem de "tıklayın" kelimeleri geçmektedir. Bu e-postanın spam olduğuna dair olasılığı kaçtır?

- A) %91
- B) %84
- C) %75
- D) %60
- E) %95

# A

## SORU 2

Bir şirketin veri tabanında A, B, C, D ve E olmak üzere beş çalışan kaydı bulunmaktadır. Her kayıt için aktif (1) veya pasif (0) durumu tutulmaktadır. Sistem yöneticisi veri tabanında tutarsızlık olduğunu fark etmiş ve kayıtları incelemiştir.

İnceleme sonucunda şu kısıtların her zaman geçerli olması gerektiği anlaşılmıştır:

Kısıt No	Kural
K1	A aktifse B de aktif olmalıdır.
K2	B aktifse C pasif olmalıdır.
K3	C veya D'den en az biri aktif olmalıdır.
K4	D aktifse E de aktif olmalıdır.
K5	E aktifse A pasif olmalıdır.

Mevcut kayıtlarda A, B, D ve E aktif; C pasiftir.

Mevcut kayıttaki kaç kısıt ihlal edilmektedir ve tutarsızlığı gidermek için hangi tek değişiklik yeterlidir?

- A) 1 kısıt ihlali var; A pasif yapılmalıdır.
- B) 2 kısıt ihlali var; E pasif yapılmalıdır.
- C) 1 kısıt ihlali var; E pasif yapılmalıdır.
- D) 2 kısıt ihlali var; A pasif yapılmalıdır.
- E) 3 kısıt ihlali var; tek değişiklikle düzeltilemez.

# A

## SORU 3

Bir güvenlik sisteminde mesajlar üç aşamalı bir dönüşüm boru hattından (pipeline) geçirilerek şifrenilmektedir. Her aşama bir fonksiyonla tanımlanmıştır:

$$f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = 2x + 3 \quad (\text{ölçekleme ve öteleme})$$

$$g: \mathbb{R} \rightarrow \mathbb{R}, g(x) = (x - 1) / 4 \quad (\text{normalleştirme})$$

$$h: \mathbb{R} \rightarrow \mathbb{R}, h(x) = x^3 + 2 \quad (\text{küpleme katmanı})$$

Şifreleme işlemi sırasıyla önce  $f$ , sonra  $g$ , son olarak  $h$  olarak uygulanır ( $h \circ g \circ f$ ). Şifresi çözülmek istenen bir mesajın şifreli değeri  $y = 10$  olarak alınmıştır. Buna göre orijinal mesaj  $x$  değeri nedir?

- A)  $x = 3$
- B)  $x = 5$
- C)  $x = 7$
- D)  $x = 9$
- E)  $x = 11$

## SORU 4

$A = \{1, 2, 3, 4\}$  kümesi üzerinde iki bağıntı tanımlanmıştır:

$$R = \{(a, b) \in A \times A \mid a + b \text{ asal sayıdır}\}$$

$$S = \{(a, b) \in A \times A \mid a \cdot b \text{ çift sayıdır}\}$$

Bileşke bağıntı  $S \circ R$  şu şekilde tanımlanır:

$$(S \circ R) = \{(a, c) \mid \exists b \in A : (a, b) \in R \text{ ve } (b, c) \in S\}$$

Buna göre,  $S \circ R$  bağıntısının eleman sayısı kaçtır ve  $(2, 2) \in S \circ R$  midir?

- A) 10 eleman,  $(2,2) \in S \circ R$
- B) 12 eleman,  $(2,2) \in S \circ R$
- C) 12 eleman,  $(2,2) \notin S \circ R$
- D) 14 eleman,  $(2,2) \in S \circ R$
- E) 16 eleman,  $(2,2) \notin S \circ R$

# A

## SORU 5

$(1 + x)^{10} \cdot (1 + x^2)^5$  açılımında  $x^4$  teriminin katsayısı kaçtır?

- A) 210
- B) 335
- C) 420
- D) 445
- E) 480

## SORU 6

Bir ağ protokolünde her paket 10 bitlik veri taşımaktadır. İletim sırasında her bit bağımsız olarak bozulabilir.  $(1+x)^{10}$  açılımındaki  $x^k$  katsayısı, tam olarak k bitinin bozulduğu farklı paket sayısını vermektedir.

Örneğin  $x^2$  katsayısı olan  $C(10,2) = 45$ , tam 2 bitin bozulduğu 45 farklı paket konfigürasyonu olduğunu ifade eder.

Protokolün hata tespit mekanizması şu özelliğe dayanmaktadır:

- Çift sayıda bozuk bit içeren paketler tespit edilemez (bu paketler geçerli görünür).
- Tek sayıda bozuk bit içeren paketler tespit edilir (bu paketler reddedilir).

Tespit edilemeyen hatalı paket sayısı ile tespit edilen hatalı paket sayısının farkı kaçtır? (Hiç hatası olmayan paket ve tüm bitleri bozulmuş paket hariç tutulacaktır.)

- A) 0
- B) 2
- C) -2
- D) 10
- E) 512

# A

## SORU 7

Bir işletim sistemi, bellek adreslerini 7 slotluk bir hash tablosuna dağıtmaktadır. 7 asal sayı olduğu için hash fonksiyonu olarak şu kullanılmaktadır:

$$h(\text{adres}) = \text{adres} \bmod 7$$

$2^{100}$  numaralı bellek adresi hash tablosunun kaçınıcı slotuna düşer?

- A) 0. slot ( $h = 0$ )
- B) 1. slot ( $h = 1$ )
- C) 2. slot ( $h = 2$ )
- D) 4. slot ( $h = 4$ )
- E) 6. slot ( $h = 6$ )

## SORU 8

$\log_2(x) + \log_2(x - 3) = 2$  denklemini sağlayan x değeri nedir?

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

# A

## SORU 9

Bir yazılım mühendisi üç farklı alt işlemten oluşan bir algoritma analiz etmektedir:

İşlem 1: 8'li arama ağacında  $n^3$  düğüm aranıyor  $\rightarrow T_1(n) = \log_8(n^3)$  adım

İşlem 2: Her adımda 4 seçenek arasından 16 tanesi değerlendiriliyor  $\rightarrow T_2 = \log_4(16)$  sabit çarpan

İşlem 3:  $n/4$  boyutlu bir veri kümesi ikiye bölünerek işleniyor  $\rightarrow T_3(n) = \log_2(n/4)$  adım

Toplam çalışma süresi aşağıdaki gibi tanımlanmıştır:

$$T(n) = T_1(n) \cdot T_2 - T_3(n)$$

Buna göre  $T(n)$  ifadesinin  $\log_2(n)$  cinsinden sadeleştirilmiş hali aşağıdakilerden hangisidir?

- A)  $\log_2(n) + 1$
- B)  $\log_2(n) + 2$
- C)  $2 \cdot \log_2(n)$
- D)  $(3/2) \cdot \log_2(n)$
- E)  $\log_2(n) + 4$

## SORU 10

$P(x) = x^3 - 6x^2 + 11x - 6$  polinomu  $(x-1)$  ile bölündüğünde kalan sıfırdır.  $P(x)$ 'in tam çarpanları hangisidir?

- A)  $(x-1)(x-2)(x-4)$
- B)  $(x-1)(x-2)(x-3)$
- C)  $(x+1)(x-2)(x-3)$
- D)  $(x-1)(x+2)(x-3)$
- E)  $(x-1)(x^2+5x+6)$

# A

## SORU 11

10x10'luk bir tahta üzerinde oynanan bir oyunda oyuncu her turda bir zar atarak gelen zar değerine göre hareket etmektedir. Oyuncu bulunduğu karede sağ, sol, yukarı veya aşağı yönlerinden birine bakıyor kabul edilir. Oyuncunun hareketi sonucunda tahtanın bir kenarından çıkması durumunda karşı kenardan tekrar tahtaya girdiği kabul edilmektedir. Örneğin, tahtanın 10. sütununda bulunan bir oyuncu 2 birim sağa gitmesi gerektiğinde tahtanın sağ kenarından çıkıp sol kenarından tahtaya girecek ve bulunduğu satırın 2. sütununa gidecektir. Tahtanın sol üst köşesinde sağa bakarak başlayan bir oyuncu, arka arkaya 4 zar atarak oynadığında tahtadaki karelerden kaç tanesine ulaşmış olabilir?

Zarların karşılıkları:

- 1: 2 kare ilerle,
- 2: saat yönünde 90 derece dön,
- 3: 1 kare geri git
- 4: saatin tersi yönünde 90 derece dön,
- 5: bekle
- 6: 1 kare ilerle

- A) 45
- B) 47
- C) 50
- D) 53
- E) 55

# A

## SORU 12

Bir öğrenci a,b,c,d,e,f,i harflerini istediği miktarda kullanarak yazabileceği tüm 5 harfli kelimelerin sayısını merak etmektedir. Bu kelimeleri türetirken öğrencinin takip etmesi gereken kurallar şunlardır:

1. kelime iki sessiz harfle başlayamaz veya bitemez
2. iki sesli harf yan yana gelemez
3. üç sessiz harf yanyana gelemez

Bu kurallar dahilinde öğrencinin yazabileceği kaç farklı kelime vardır?

- A) 2016
- B) 2160
- C) 1872
- D) 1728
- E) 2304

# A

## SORU 13

Bir öğrenci aşağıda verilen algoritmayı bir dairesel array üzerinde çalıştırmak istemektedir.  $k$  elemanlı dairesel bir array'de  $i \leq k$  olmak kaydıyla  $k+i$ 'inci eleman  $i$ 'inci eleman kabul edilir. Array başlangıçta 8 büyüklüğünde olup, içinde 1'den 8'e kadar sırayla sayılar tutulmaktadır. *pointer* başlangıçta 1. elemana işaret etmekte olup, *pointer*'da tutulan değer 9'dur (1. eleman arraydeki ilk elemandır).

Algoritma'da kullanılan yardımcı işlemler:

- `ilerle(x)`: *pointer*'ı  $x$  eleman ilerlet (saat yönünde).
- `degistir()`: *pointer*'ın işaret ettiği elemanın değeriyle *pointer*'da tutulan değeri birbiriyle değiştir.
- `p_deger()`: *pointer*'da tutulan değeri dön

1. `p_deger()` tek mi?

1.1. evet:

1.1.1. `degistir()`

1.1.2. `ilerle(p_deger())`

1.2. hayır:

1.2.1. `ilerle(p_deger() / 2)`

1.2.2. `degistir()`

2. `p_deger() == 9` ise dur; değilse 1.'e git

Algoritma çalıştırılıp durduğunda arrayin son hali hangisinde verildiği gibi olur?

A) sonsuza kadar çalışmaya devam eder.

B) [1,2,3,4,5,6,7,8]

C) [6,1,3,8,5,4,7,2]

D) [6,1,3,2,5,4,7,8]

E) [6,1,2,4,5,3,7,8]

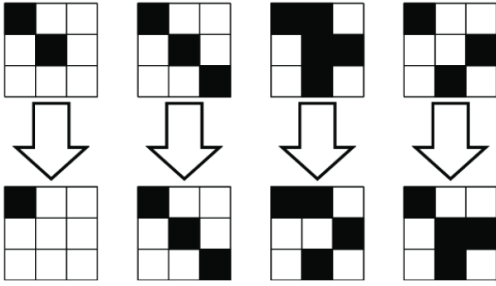
# A

## SORU 14

Bir kenarı 1 birim olan bir kare 4 özdeş parçaya bölündüğünde her bir parçanın çevresi en fazla kaç birim olabilir?

- A) 2
- B) 2.5
- C) 4
- D) 6.32
- E) sonsuz

## SORU 15



Conway'in "hayat oyunu"nda, eş kare hücrelere bölünmüş (grid) sonsuz büyüklükteki bir tahtada temel birkaç kural yardımı ile basit bir hayat simülasyonu yapılır. Her bir hücre ya boştur, ya da içinde bir canlı organizma yaşamaktadır. Oyun tur bazlıdır, her turda kurallar tüm kareler için hesaplanır ve bir sonraki tura geçilir. Tüm kurallar tüm karelere eş zamanlı olarak uygulanır. Kurallar şu şekildedir:

- 1 veya 0 komşusu canlı olan karelerdeki canlılar ölür
- 2 veya 3 komşusu canlı olan karelerdeki canlılar yaşamaya devam eder
- 4 veya daha fazla komşusu canlı olan karelerdeki canlılar ölür
- 3 komşusu olan her boş karede yeni bir canlı yaşamaya başlar

Aşağıdaki başlangıçlardan hangisi sonsuza kadar yaşayan bir sistem üretir? (koyu renk kareler canlı karelerdir) (tahtada en az bir canlı kare bulunması yaşamın devam ettiği anlamına gelir)

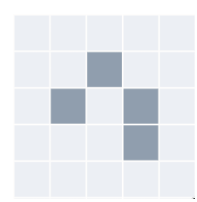
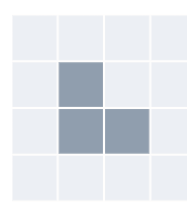
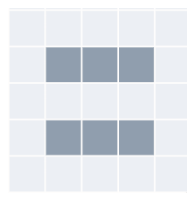
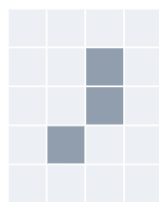
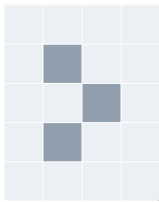
A)

B)

C)

D)

E)



# A

## [Soru 16-17 için açıklama]

Bir tamsayı dizisini soldan sağa işleyen bir yığın algoritması tanımlayalım. Yığın başlangıçta boşdur. Dizinin elemanları sırayla alınır ve her eleman  $x$  için aşağıdaki kurallar uygulanır.

- Yığın boş ise  $x$  yığığa eklenir.
- Yığın boş değilse ve yığının üstündeki eleman  $x$ 'e eşitse, üstteki eleman silinir.
- Aksi halde, yığının üstündeki eleman **üstteki eleman**  $- x$  değeri ile değiştirilir.

Dizinin tüm elemanları işlendiğinde algoritma sonlanır.

Örnek olarak **{5, 1, 4}** dizisi işlendiğinde yığın sırasıyla **{5}**, sonra **{4}**, sonra **{}** olur.

### SORU 16

A: {7, 3, 10, 6, 4, 2, 8} olduğunda algoritma sonlandığında yığının üstündeki eleman ne olur?

- A) 2
- B) 4
- C) 6
- D) 8
- E) 10

### SORU 17

A: {4, 9, 5, 1, 8, 2, 6} dizisinin sonuna tek bir sayı eklenecektir. Aşağıdakilerden hangisi eklenirse algoritma sonlandığında yığın boş olur?

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

# A

## [Soru 18-19 için açıklama]

Bir bilgisayar oyununda ekran üzerinde yan yana dizilmiş  $n$  adet sayı blokundan oluşan bir dizi bulunmaktadır. Oyuncu, her hamlede yan yana duran ve değerleri birbirine eşit olan iki blok seçer; bu iki blok silinerek yerlerine değerlerinin 1 fazlası olan yeni bir blok eklenir. Sağda kalan bloklar sola kayarak dizinin bütünlüğü korunur.

Oyuncu, yan yana duran eşit değerli blok kalmayana kadar bu işlemi tekrarlar. Birden fazla birleştirme seçeneği olması durumunda, oyuncu istediği çifti seçmekte özgürdür.

Örnek olarak başlangıç dizisi  $A = [2, 1, 1, 2]$  olsun:

- Oyuncu 1 ve 1'i birleştirirse dizi  $[2, 2, 2]$  olur. Ardından 2'leri birleştirerek  $[3, 2]$  veya  $[2, 3]$  elde eder (kalan eleman sayısı 2).
- Eğer dizi  $A = [1, 1, 1, 1]$  olsaydı, sırasıyla  $[2, 1, 1]$ ,  $[2, 2]$  ve nihayetinde  $[3]$  elde edilirdi (kalan eleman sayısı 1).

### SORU 18

Başlangıç dizisi  $A = [3, 2, 2, 1, 1, 2, 3, 4]$  olarak verildiğinde, en zekice seçimler yapıldığında oyun bittiğinde dizide en az kaç eleman kalır?

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

### SORU 19

Başlangıç dizisi  $A = [3, 2, 1, 1, 1, 1, 1, 1, 2, 3]$  olarak verildiğinde, en kötü senaryoda (rastgele ve hatalı seçimler sonucunda) oyun tıkanıldığında dizide en fazla kaç eleman kalabilir?

- A) 6
- B) 7
- C) 8
- D) 9
- E) 10

# A

## SORU 20

Bir depo otomasyon sisteminde görevli robot kol, yan yana dizilmiş  $n$  adet farklı ağırlıktaki koliyi hafiften ağıra doğru (soldan sağa) sıralama ile yükümlüdür. Robotun icra edebildiği yegane işlem, dizinin en solundan başlayarak belirlenen bir  $k$ . sıradaki koliye kadar olan bölümü tersyüz etmektir.

Robotun sıralama algoritması adım adım şu kuralları uygular:

1. Sıralanmamış kısımdaki en ağır koliyi tespit et.
2. Eğer bu koli halihazırda sıralanmamış kısmın en sağında ise 5. adıma geç.
3. Eğer koli en sol (1. sıra) konumda değilse, kolinin bulunduğu yere kadar olan alt diziyi ters çevirerek en başa getir.
4. En sol tarafa gelmiş olan koliyi, sıralanmamış kısmın sonuna yerleştirmek için ilgili alt diziyi ters çevir.
5. Sıralanmış kabul edilen koli sayısını bir artır ve kalan koliler için 1. adıma dön.

Başlangıç ağırlık dizisi  $A = [12, 5, 18, 4, 15, 8, 10]$  olarak verilmiştir.

En ağır **iki** koli kendi nihai yerlerine yerleştirildiği anda (üçüncü döngüye geçilmeden hemen önce), dizide baştan (soldan) 2. sırada hangi koli bulunur?

- A) 5
- B) 8
- C) 10
- D) 12
- E) 15

## SORU 21

8 farklı kitap bir rafa dizilecektir. Ancak belirli 3 kitap yan yana gelmemelidir. Kaç farklı diziliş vardır?

- A)  $8! - 3! \cdot 6!$
- B)  $8! - 3! \cdot 6! \cdot 2$
- C)  $8! - 6!$
- D)  $8! - 3! \cdot 5!$
- E) Hiçbiri

# A

## SORU 22

10 kişilik bir gruptan 4 kişilik bir komite seçilecektir. Ancak bu gruptaki iki kişi A ve B aynı anda komitede bulunamaz. Kaç farklı seçim yapılabilir?

- A) 70
- B) 140
- C) 154
- D) 182
- E) 308

## SORU 23

Bir üniversite senatosunda 6 Profesör ve 8 Doçent bulunmaktadır. Bu gruptan 5 kişilik bir araştırma komisyonu seçilecektir. Ancak seçim için şu kısıtlamalar getirilmiştir:

- Komisyonda en az 2 Profesör bulunmak zorundadır.
- Profesörler arasındaki A ve B şahısları, aynı komisyonda beraber yer alamazlar.

Bu koşullara uygun kaç farklı komisyon kurulabilir?

- A) 1120
- B) 1306
- C) 1406
- D) 1526
- E) 1746

## SORU 24

5 Elma ve 5 portakal bir sıraya dizilecektir. Elmalar ve portakallar alternatif dizilecektir. Yani sıranın bir yerinde elma varsa sonra portakal sonra elma vs. şeklinde bir dizilim olacaktır. Kaç farklı dizilim mümkündür?

- A) 252
- B) 14400
- C) 28800
- D) 3628800
- E) Hiçbiri

# A

## SORU 25

1'den 200'e kadar olan (1 ve 200 dahil) doğal sayılar arasından kaç tanesi 2, 3 ve 5 sayılarının hiçbirine bölünmez?

- A) 52
- B) 53
- C) 54
- D) 55
- E) 56

## SORU 26

Yönlü ve döngüsüz bir çizge (DAG) için aşağıdakilerden hangisi topolojik sıralamanın tek (unique) olması için gerekli ve yeterlidir?

- A) Çizge bağlıdır
- B) Her düğüme gelen kenar sayısı (indegree)  $\leq 1$ 'dir
- C) Her ardışık düğüm çifti arasında bir kenar vardır
- D) Çizge bir ağaçtır
- E) Tüm düğümlerden çıkan kenar sayısı (outdegree)  $\geq 1$ 'dir

## SORU 27

Genişlik öncelikli arama (Breadth First Search - BFS) ve Derinlik öncelikli arama (Depth First Search - DFS) ile ilgili aşağıdakilerden hangisi her zaman doğrudur?

- A) DFS ağacı minimum yüksekliktedir.
- B) BFS ağacı minimum yüksekliktedir.
- C) DFS aynı zamanda başlangıç düğümünden diğer düğümlere olan en kısa yolu bulur.
- D) BFS ağırlıklı çizgelere uygulandığında aynı zamanda başlangıç düğümünden diğer düğümlere olan ağırlıklı en kısa yolu bulur.
- E) DFS her zaman BFS'ten daha hızlıdır.

# A

## SORU 28

Bir yönlü çizge veriliyor. Bu çizgenin güçlü bağlı bileşenleri (SCC) bulunuyor ve her bir SCC, tek bir düğüm olacak şekilde sıkıştırılıyor. Bu şekilde elde edilen yeni çizgeye yoğunlaşım çizgesi denir.

Yoğunlaşım çizgesinde kenarlar şu şekilde oluşturulur: Orijinal çizgede farklı iki SCC arasında, birinciden ikinciye yönlü bir kenar varsa, yoğunlaşım çizgesinde de bu iki SCC'ye karşılık gelen düğümler arasında yönlü bir kenar bulunur.

Bu yoğunlaşım çizgesi için aşağıdakilerden hangisi her zaman doğrudur?

- A) Her zaman bağlıdır.
- B) Her zaman iki bölümlü (bipartite)'tir.
- C) Her zaman yönsüzdür.
- D) Hiçbir zaman döngü içermez.
- E) Her zaman ağaçtır.

## SORU 29

Bir çizgede,  $V$  düğüm ve  $E$  kenar sayısını temsil etmek üzere "Seyrek (sparse)" çizgelerde kenar sayısı düğüm sayısına yakındır (yani çok fazla kenar yoktur).

Bir çizgeyi temsil etmek için iki yöntem vardır:

Komşuluk Matrisi (Adjacency Matrix)  
Komşuluk Listesi (Adjacency List)

Seyrek bir çizgede, Komşuluk Matrisi yerine Komşuluk Listesi kullanmanın en önemli avantajı aşağıdakilerden hangisidir?

- A) İki düğüm arasında kenar olup olmadığını her zaman sabit sürede kontrol edebilmek
- B) Kullanılan belleği, düğüm sayısının karesi kadar yer kaplamak yerine, yalnızca düğüm ve kenar sayısı kadar olacak şekilde azaltmak
- C) Bir düğümün kaç komşusu olduğunu bulmak için, sadece o düğüme ait küçük bir listeyi incelemek yerine tüm düğümler üzerinden tek tek kontrol yapmak zorunda kalmak
- D) Yeni bir kenar eklerken çok daha fazla işlem yapmak zorunda kalmak
- E) Çok sayıda kenarı olan yoğun çizgelerde daha hızlı çalışmak

# A

## SORU 30

Aşağıdaki yönlü çizgenin komşuluk listeleri aşağıda verilen sırayla gezilmektedir ve Derinlik Öncelikli Arama (Depth First Search - DFS), düğüm A'dan başlatılmaktadır.

- A: B, C
- B: D, E
- C: F
- D: C, F
- E: F, G
- F: H
- G: D, H
- H: —

DFS sırasında bir düğüm ilk kez görüldüğünde ona bir keşif zamanı  $d[u]$ , o düğümün tüm komşuları tamamlandığında ise bir bitiş zamanı  $f[u]$  veriliyor. Sayaç her keşif ve her bitişte 1 artıyor; başlangıçta sayacın değeri 0'dır.

Buna göre aşağıdakilerden hangisi doğrudur?

- A)  $d[F]=6$  ve  $f[F]=9$
- B)  $d[C]=4$  ve  $f[C]=7$
- C)  $d[E]=10$  ve  $f[E]=15$
- D)  $d[G]=11$  ve  $f[G]=14$
- E)  $d[D]=3$  ve  $f[D]=10$

# A

## SORU 31

Kenar ağırlıkları pozitif olan bir çizge verilmiştir. Her düğüm (ing: vertex) bir şehri, her kenar (ing: edge) ise iki şehir arasındaki kara yolu mesafesini temsil etmektedir. Bir  $s$  şehriden  $t$  şehrine gitmek için aşağıdaki arama yöntemi kullanılmaktadır:

- Arama yöntemi: Derinlik Öncelikli Arama (ing: Depth-First Search, DFS)
- Her düğüm için bir  $h(n)$  değeri tanımlıdır.
- $h(n)$  düğüm  $n$ 'den hedef  $t$ 'ye olan kuş uçuşu mesafe tahminini verir; bu nedenle  $h(t) = 0$  dır.
- Bir düğüm genişletildiğinde, komşular küçük  $h(n)$  değerli olan önce ziyaret edilecek şekilde önceliklendirilir.
- DFS yığını (ing: stack) son-giren-ilk-çıkarm (ing: Last-in-first-out, LIFO) olduğu için, bu önceliği sağlamak amacıyla komşular yığına büyük  $h(n)$ 'den küçük  $h(n)$ 'e doğru eklenir.
- Algoritma hedef düğüm  $t$ 'ye ulaştığı anda durur ve bulduğu yolu döner.

*En kısa yol*, toplam kenar maliyeti en küçük yol anlamındadır. Yukarıda tarif edilen bu algoritma ile ilgili aşağıdakilerden hangisi doğrudur?

- A) Kuş uçuşu mesafe kullanıldığı için algoritma her zaman en kısa yolu bulur.
- B) Algoritma, tüm yolları sistematik olarak denediği için her zaman en iyi çözümü bulur.
- C)  $h(n)$  kullanımı algoritmanın davranışını değiştirmez, sonuçlar standart DFS ile aynıdır.
- D) Algoritma, Genişlik Öncelikli Arama (ing: Breadth-First Search, BFS) gibi çalıştığı için en kısa yolu bulur.
- E) Algoritma, standart DFS'e göre genellikle daha hızlı sonuç verebilir ve bazı durumlarda daha iyi (daha kısa) bir yol bulabilir, ancak en kısa yolu garanti etmez.

# A

## [Soru 32-50 için açıklama]

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

### SORU 32

Kenar uzunluğu 2 olan bir kare, koordinat düzleminde orijin merkezlidir. Bu durumda bu karenin köşeleri  $(-1, -1)$ ,  $(-1, 1)$ ,  $(1, -1)$ ,  $(1, 1)$  noktalarındadır. Bu karenin içinde, yarıçapı 1 olan ve merkezi yine orijinde bulunan bir çember yer almaktadır. Aşağıdaki C programlama dilinde yazılmış olan bilgisayar programı  $N$  defa işleyen bir döngü içinde (a) kare içinde eşit olasılıkla rastgele bir konumda bir nokta üretir, (b) bu noktalardan kaç tanesinin çemberin içinde kaldığını  $M$  tamsayı değişkeni içinde sayar. Program döngü bittikten sonra  $(4.0 * M) / N$  değerini döner. *İpucu:*  $M / N$  oranı üzerinden düşününüz.

```
double hesapla(int N) {
    int M = 0;
    for (int i = 0; i < N; i++) {
        double x = -1.0 + 2.0 * rand() / (double)RAND_MAX;
        double y = -1.0 + 2.0 * rand() / (double)RAND_MAX;
        if (x * x + y * y <= 1.0) M++;
    }
    return (4.0 * M) / N;
}
```

$N$  çok büyük olduğunda (yani  $N \rightarrow \infty$ ), fonksiyonun döndüğü değer aşağıdakilerden hangisine en yakındır?

**Not:** `rand()` fonksiyonu 0 ile `RAND_MAX` arasında rastgele bir tamsayı üretir. `RAND_MAX`, `rand()` fonksiyonunun üretebileceği en büyük değeri temsil eder.

- A)  $\pi/4$
- B) 2
- C) 3
- D)  $\pi$
- E) 4

# A

## SORU 33

Bir merdivende 0'dan  $n$ 'e kadar (0 ve  $n$  dahil) numaralandırılmış basamaklar vardır. Verilen  $c$  dizisi için  $c[i]$  değeri  $i$  numaralı basamağa basmanın maliyetini göstermektedir. Bir kişi ilk başta 0. basamakta sıfır maliyet ile durmaktadır. Bu kişi her hareketinde yalnızca bir basamak veya iki basamak çıkabilmektedir.  $F(i)$  ile  $i$ . basamağa en az toplam maliyetle ulaşma değerini aşağıdaki şekilde tanımlanmıştır:

$$F(0) = 0, \quad F(1) = c[1], \quad F(i) = \min(F(i - 1), F(i - 2)) + c[i] \quad (i \geq 2 \text{ için})$$

Bu bağıntıyı verimli bir şekilde hesaplamak için aşağıdaki kod kullanılıyor:

```
#define ATANMAMIS -1
int F(int i, int c[], int memo[]) {
    if (i == 0) return 0;
    if (i == 1) return c[1];
    if (memo[i] == ATANMAMIS) {
        int a = F(i - 1, c, memo);
        int b = F(i - 2, c, memo);
        memo[i] = min(a, b) + c[i];
    }
    return memo[i];
}
```

Başlangıçta memo dizisindeki tüm elemanlar ATANMAMIS değerine ayarlanmıştır.  $A$  ve  $B$ ,  $n$ 'den bağımsız tamsayı sabit iki değer olsun.  $F(n, c, memo)$  çağrısı yapıldığında, memo dolu olup hemen dönen çağrılar da dahil olmak üzere,  $F(n, c, memo)$  fonksiyonu toplam kaç defa çağrılmaktadır?

**Not:** Bu koddaki  $\min(a, b)$  fonksiyonu verilen  $a$  ve  $b$  parametrelerinden minimumunu dönmek üzere tanımlandığını kabul edin.

- A)  $A$
- B)  $A \log(n) + B$
- C)  $An + B$
- D)  $An^2 + B$
- E)  $Af^n + B$  ( $1 < f \leq 2$ )

# A

## SORU 34

```
struct YonelimliCevreleyenKutu {  
    double merkezX;  
    double merkezY;  
    float aci;  
    unsigned short genislik;    // desimetre  
    unsigned short yukseklik;  // desimetre  
};
```

### Varsayımlar

- double: 8 byte
- float: 4 byte
- unsigned short: 2 byte
- hizalama: 8 byte
- sistem: 64-bit

Yukarıda verilen yapı için aşağıdaki ifadelerden hangisi tamamen doğrudur?

(1 metre = 10 desimetre, padding = dolgu, alignment = hizalama)

- A) Toplam boyut 24 byte'tır. Dolgu yoktur. `genislik` en fazla 6553.5 metre tutabilir.
- B) Toplam boyut 24 byte'tır. Dolgu yoktur. `genislik` en fazla 65535 metre tutabilir.
- C) Toplam boyut 28 byte'tır ve dolgu yoktur. `genislik` en fazla 6553.5 metre tutabilir.
- D) Toplam boyut 32 byte'tır. Sonda 4 byte dolgu vardır. `genislik` en fazla 6553.5 metre tutabilir.
- E) Toplam boyut 32 byte'tır. Sonda 8 byte dolgu vardır. `genislik` en fazla 6553.5 metre tutabilir.

## SORU 35

Aşağıdaki fonksiyonun işlevi nedir?

```
int bitUzerindeIslemYap(int x) {  
    return x & (x - 1);  
}
```

- A) Sayının sağdan itibaren ilk 1 değerindeki bitini izole eder (diğer tüm bitleri 0 yapar)
- B) Sayının sağdan itibaren ilk 1 değerindeki bitini 0 yapar
- C) Sayının tek olup olmadığını kontrol eder
- D) Sayının tüm bitlerini ters çevirir
- E) Sayıyı iki katına çıkarır

# A

## SORU 36

```
int main() {  
    int i, j;  
    for (i = 0; i < 4; i++) {  
        for (j = i; j < 5; j++) {  
            if ((i + j) % 2 == 0)  
                printf("*");  
        }  
    }  
    return 0;  
}
```

Yukarıdaki program ekrana kaç tane yıldız "\*" basar?

- A) 5
- B) 6
- C) 7
- D) 8
- E) 9

# A

## SORU 37

```
int temp() {
    int a[5] = {1, 3, 5, 7, 9};
    int *p = a;

    int t = 0;
    int i;
    for (i = 0; i < 5; i += ?) {
        t += *(p + i);
    }

    return t;
}
```

Yukarıdaki fonksiyonun 15 değerini dönmesi için for döngüsü içindeki ? yerine aşağıdakilerden hangisi yazılmalıdır?

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

## SORU 38

```
int f(int n) {
    if (n <= 1) return 1;
    return f(n - 1) + f(n - 2);
}

int main() {
    printf("%d\n", f(5));
    return 0;
}
```

Yukarıdaki kod hangi çıktıyı üretir?

- A) 5
- B) 8
- C) 13
- D) 15
- E) Hiçbiri

# A

## SORU 39

```
void f(int *x, int n) {
    if (n == 0) return;
    *x += n;
    f(x, n - 1);
    *x -= 1;
}

int main() {
    int x = 0;
    f(&x, 4);
    printf("%d\n", x);
    return 0;
}
```

Yukarıdaki kod hangi çıktıyı üretir?

- A) 6
- B) 8
- C) 12
- D) 0
- E) Program hata vereceğinden çıktı oluşmaz

## SORU 40

```
void foo(int n) {
    if (n <= 0) return;
    foo(n - 1);
    printf("*");
    foo(n - 1);
}
```

Yukarıdaki fonksiyon `foo(3)` ; şeklinde çağrılırsa ekrana kaç tane yıldız "\*" yazdırılır?

- A) 3
- B) 5
- C) 7
- D) 8
- E) 15

# A

## SORU 41

```
int main() {
    char s[] = "OLIMPIYAT";
    int i, j;

    for (i = 0, j = strlen(s) - 1; i < j; i += 2, j -= 3) {
        char t = s[i];
        s[i] = s[j];
        s[j] = t;
    }

    for (i = 0; s[i] != '\0'; i++) {
        if (s[i] == 'I' || s[i] == 'A')
            s[i] = '*';
    }

    printf("%s", s);
    return 0;
}
```

Yukarıdaki kodun çıktısı nedir?

- A) TL\*MP\*Y\*O
- B) TL\*MPY\*O\*
- C) TO\*MP\*Y\*L
- D) TL\*MPIYAO
- E) OL\*MP\*Y\*T

# A

## SORU 42

```
void f(int n) {
    printf("*");
    if (n <= 1) return;
    f(n / 2);
    if (n % 2 == 0)
        f(n / 2 - 1);
}

int main() {
    f(10);
    return 0;
}
```

Yukarıdaki kod çalıştığında ekrana kaç adet yıldız "\*" karakteri yazdırılır?

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

# A

## SORU 43

```
int main() {
    int A[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;

    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            if (i < j)
                A[j][i] = A[i][j] - A[j][i];
        }
    }

    for (i = 0; i < 3; i++)
        printf("%d ", A[i][0] + A[i][1] + A[i][2]);

    return 0;
}
```

Yukarıdaki kodun çıktısı nedir?

- A) 6 9 3
- B) 6 3 9
- C) 9 6 3
- D) 3 9 6
- E) 6 9 6

# A

## SORU 44

```
int f(int n) {  
    int s = 0;  
    while (n > 0) {  
        s += n & 1;  
        n >>= 1;  
    }  
    return s;  
}
```

Yukarıdaki  $f$  fonksiyonunun amacı ne olabilir?

- A)  $n$  sayısının onluk tabandaki basamak sayısını hesaplamak
- B)  $n$  sayısının ikilik sistem gösterimindeki 1 bitlerinin sayısını hesaplamak
- C)  $n$  sayısının en büyük asal bölenini bulmak
- D)  $n$  sayısının tek olup olmadığını belirlemek
- E)  $n$  sayısının 2'nin kuvveti olup olmadığını kontrol etmek

# A

## SORU 45

```
int main() {
    int a[] = {2, 4, 6, 8, 10};
    int *p = a + 4;

    while (p > a) {
        *(p - 1) = *p - *(p - 1);
        p--;
    }

    return 0;
}
```

Yukarıdaki kodun çalıştırılması neticesinde aşağıdakilerden hangisi doğru olur?

- A) a[0] değeri -8 olur.
- B) a[4] değeri 2 olur.
- C) Dizi elemanlarının toplamı -10 olur.
- D) Dizide negatif eleman kalmaz.
- E) a[2] değeri -6 olur.

# A

## SORU 46

```
int main() {
    int arr[10] = {2, 3, 5, 1, 9, 2, 8, 4, 0, 7};

    int i = 0;
    int t = 0;

    do {
        int g = arr[i];

        t += g;

        arr[i] = (arr[i] + 1) % 10;
        i = g;

    } while (i != 0);

    printf("%d\n", t);
    return 0;
}
```

Yukarıdaki program çalıştırıldığında ekrana hangi çıktıyı yazdırır?

- A) 15
- B) 17
- C) 23
- D) 29
- E) 41

# A

## SORU 47

```
void islem(int *arr, int n) {
    if (n <= 1) {
        return;
    }
    islem(arr + 1, n - 1);

    arr[0] = arr[0] + arr[1];
}

int main() {
    int dizi[] = {3, 6, 5, 2, 4};
    islem(dizi, 5);

    for(int i = 0; i < 5; i++) {
        printf("%d ", dizi[i]);
    }
    return 0;
}
```

Yukarıdaki program çalıştırıldığında, main fonksiyonundaki döngü ekrana hangi çıktıyı yazdırır?

- A) 20 17 11 6 4
- B) 9 11 7 6 4
- C) 3 9 14 16 20
- D) 16 13 7 2 4
- E) 3 6 5 2 4

# A

## SORU 48

```
int f1(int x, int y) {
    if (x == 0 || y == 0) {
        return 1;
    }
    return f1(x - 1, y) + f1(x, y - 1) + f1(x - 1, y - 1);
}

int main() {
    printf("%d\n", f1(2, 3));
    return 0;
}
```

Yukarıdaki program çalıştırıldığında, ekrana hangi çıktıyı yazdırır?

- A) 10
- B) 13
- C) 20
- D) 25
- E) 31

# A

## SORU 49

```
int f(int n, int c, int a, int b) {
    if (a < b) {
        return c;
    }

    if ((c & 1) == 0) {
        int x = (n & b) ? 1 : 0;

        return f(n, (c << 1) | x, a, b << 1);
    } else {
        int x = (n & a) ? 1 : 0;

        return f(n, (c << 1) | x, a >> 1, b);
    }
}
```

Yukarıdaki program çalıştırıldığında, ekrana hangi çıktıyı yazdırır?

- A) 14
- B) 19
- C) 22
- D) 25
- E) 28

# A

## SORU 50

```
void degistir(int x, int y){  
    x = x op1 y;  
    y = x op2 y;  
    x = x op3 y;  
    printf("%d %d\n", x, y);  
}
```

Yukarıdaki fonksiyon, harici bir geçici değişken kullanmadan x ve y parametrelerinin değerlerini kendi aralarında değiştirmeyi (swap) amaçlamaktadır. Gönderilecek x ve y değerlerinin sadece rakamlar olduğu bilinmektedir. Buna göre, fonksiyonun bu değiştirme işlemini her koşulda doğru bir şekilde gerçekleştirmesi için koddaki op1, op2 ve op3 yerlerine sırasıyla aşağıdaki operatör gruplarından hangisi getirilmelidir?

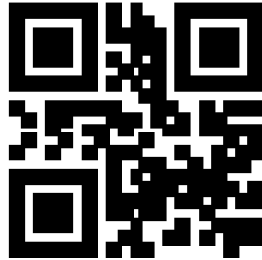
- I. +, -, -
- II. ^, ^, ^
- III. \*, /, \*
- IV. -, +, -

- A) I ve II
- B) I ve III
- C) II ve III
- D) I, II ve III
- E) I, II ve IV

**SINAV BİTTİ**  
Cevaplarınızı kontrol ediniz.

A

BU SAYFA  
BOŞ BIRAKILMIŞTIR.



Lise Bilgisayar (A)			
1	E	26	C
2	A	27	B
3	A	28	D
4	B	29	B
5	D	30	E
6	C	31	E
7	C	32	D
8	D	33	C
9	B	34	A
10	B	35	B
11	A	36	D
12	B	37	C
13	C	38	B
14	E	39	A
15	D	40	C
16	C	41	A
17	A	42	C
18	B	43	A
19	C	44	B
20	D	45	C
21	A	46	C
22	D	47	A
23	B	48	D
24	C	49	E
25	C	50	A

Lise Bilgisayar (B)			
1	D	26	C
2	B	27	B
3	C	28	D
4	C	29	B
5	E	30	E
6	D	31	E
7	C	32	D
8	A	33	A
9	B	34	C
10	C	35	A
11	A	36	B
12	D	37	A
13	C	38	D
14	D	39	E
15	E	40	A
16	A	41	C
17	A	42	A
18	B	43	B
19	D	44	D
20	C	45	C
21	B	46	B
22	B	47	A
23	A	48	C
24	B	49	C
25	C	50	C