Sevgi’nin Koleksiyonu Çözüm

Bu soruda yapabileceğimiz gözlemlerden birisi durumların birbirinden ayrık olmasıdır. Yani bir elemanı göz önünde bulundurduğumuzda onu etkileyen yalnızca ondan önceki ve sonraki eleman olabilir. Bu sayede Dinamik Programlama’da kolaylıkla ifade edebileceğimiz durumlara dönüştürebiliriz.

Dinamik yapısını kurarken her seferinde kaçıncı indexte olduğumuzu ve benden öncekini 2. Diziden alıp almama durumuna göre 1/0 değeri. Yani 2*N maliyetle bu yapıyı kurabiliyoruz.

Bu sayede her bir dinamik durumunda eğer gelen değer 0 ise benden önceki 1. Dizideki elemana göre işlem yapıyorken 1 ise 2. Diziye göre işlem yapıyorum.

Dinamik dizimin içinde ise o zamana kadar kaç adet 2. Diziden eleman ödünç aldığımı tutarak minimumunu hesaplıyorum.

Çözüm Kodu:

```
#include<bits/stdc++.h>

using namespace std;

int n,a[100005],b[100005];
int dp[100005][2];

int f (int x,int aldimMi) {
	if (x == n+1)
		return 0;

	if (dp[x][aldimMi] != -1)
		return dp[x][aldimMi];

	dp[x][aldimMi] = 100000000;

	int prev = aldimMi ? b[x-1] : a[x-1];

	if (prev >= a[x])
		dp[x][aldimMi] = f(x+1,0);

	if (prev >= b[x])
		dp[x][aldimMi] = min(dp[x][aldimMi], f(x+1,1)+1);
```

```
	return dp[x][aldimMi];
}

int main(){
	cin >> n;
	for (int i=1;i<=n;i++) cin >> a[i];
	for (int i=1;i<=n;i++) cin >> b[i];

	memset(dp,-1,sizeof dp);
	int ans = min(f(2,0), f(2,1)+1);
	if (ans >= 100000000)
		puts("-1");
	else
		cout << ans << endl;
}
```