

| Kapasite $\rightarrow$ |       |         | $j = 0$ | 1 | 2 | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|------------------------|-------|---------|---------|---|---|----|----|----|----|----|----|----|----|
| Eleman                 | Değer | Ağırlık | 0       | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 1                      | 10    | 5       | 0       | 0 | 0 | 0  | 0  | 10 | 10 | 10 | 10 | 10 | 10 |
| 2                      | 40    | 4       | 0       | 0 | 0 | 0  | 40 | 40 | 40 | 40 | 40 | 50 | 50 |
| 3                      | 30    | 6       | 0       | 0 | 0 | 0  | 40 | 40 | 40 | 40 | 40 | 50 | 70 |
| 4                      | 50    | 3       | 0       | 0 | 0 | 50 | 50 | 50 | 50 | 90 | 90 | 90 | 90 |

Sırt çantası problemi dinamik programlama yaklaşımı ile çözülmüş ve aşağıdaki hafıza tablosu elde edilmiştir. Bu tabloya göre çantada hangi elemanlar olmalıdır?

- A** ☐ 2 ve 4
- B** ☐ 1 ve 2
- C** ☐ 2 ve 3
- D** ☐ 3 ve 5
- E** ☐ 4 ve 5

m ve n elemanlı iki sıralı dizi veriliyor ve etkin bir böl yönet stratejisi uygulanarak, iki dizinin bitişik halinde k. elemanı aranmak isteniyor. Maliyet nedir?

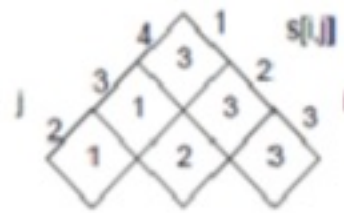
**A** ☐  $O(m+n)$

**B** ☐  $O(n \log n)$

**C** ☐  $O(\log(\max(m,n)))$

**D** ☐  $O(\log n)$

**E** ☐  $O(1)$



A1, A2, A3 ve A4 matrislerinin çarpımı için dinamik programlama yöntemi uygulanmış ve Matrislerin çarpım sırasına ilişkin bilgiyi içeren tablo (memoize) aşağıdaki gibi oluşmuştur. Buna göre matrislerin çarpım düzeni nasıl olmalıdır?

- A** ☐  $\{((A1.A2).A3).A4\}$
- B** ☐  $\{(A1.A2).(A3.A4)\}$
- C** ☐  $\{A1.(A2.(A3.A4))\}$
- D** ☐  $\{A1.((A2.A3).A4)\}$
- E** ☐  $\{(A1.(A2.A3)).A4\}$

Rekürsif tanımı  $T(1)=1$  ve  $n \geq 2$  için  $T(n)=T(n/2) + 6n - 1$  biçiminde verilen algoritmanın büyüme hızı nedir?

- A** ☐  $3n^2 + 6n + 1$
- B** ☐  $8n^2$
- C** ☐  $\log \log n + 0.25n$
- D** ☐  $6n \log n + 1$
- E** ☐  $3n^2 \log n + 6n$

Seçimi Boş Bırakmak İstiyorum

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 4T(n/2) + n & \text{otherwise.} \end{cases}$$

Şekilde rekürsif tanımı verilen algoritmanın çalışma zamanının asimptotik ifadesi hangisidir?

- A** ☐  $O(n^2 \log n)$  (2 üs olarak değerlendirilecek, karesel)
- B** ☐  $O(n \log^2 n)$  (2 üs olarak değerlendirilecek, log kare n)
- C** ☐  $O(n \log n)$
- D** ☐  $O(n)$
- E** ☐  $O(n^2)$  (2, üs olarak değerlendirilecek, karesel)

Seçimi Boş Bırakmak İstiyorum

$$x^{30}$$

$x^{30}$  sayısının hesaplanması için Transform&Conquer yöntemi ile tasarlanmış *Right-to-left binary exponentiation algoritması* kullanılırsa aşağıdaki sonuçlardan hangisi doğrudur?

- A** ☐ Bu değer hesabı 5 adımda hesaplanabilir.
- B** ☐ Bu değer hesabı 30 adımda hesaplanabilir.
- C** ☐ Bu değer 900 adımda hesaplanır
- D** ☐ Bu değer  $30 \times 4 = 120$  adımda hesaplanır
- E** ☐ Bu değer üs ten bağımsız olarak 1 saniyede hesaplanır.

Seçimi Boş Bırakmak İstiyorum

Dinamik programlama yaklaşımı ile geliştirilen zincir matris çarpımı algoritması ile ilgili olarak aşağıdakilerden hangisi yanlıştır?

- A** ☐ Rekürsiyonun maliyeti minimuma indirilmiştir.
- B** ☐ Minimum Çarpma sayısını verirken çarpma düzeni hakkında bilgi içermez.
- C** ☐ Bu problem Brute Force tekniği ile üstel bir maliyetle çözülür
- D** ☐ Algoritmanın zaman karmaşıklığı  $O(n^3)$
- E** ☐ Algoritmanın bellek zaman karmaşıklığı  $O(n^2)$

Seçimi Boş Bırakmak İstiyorum

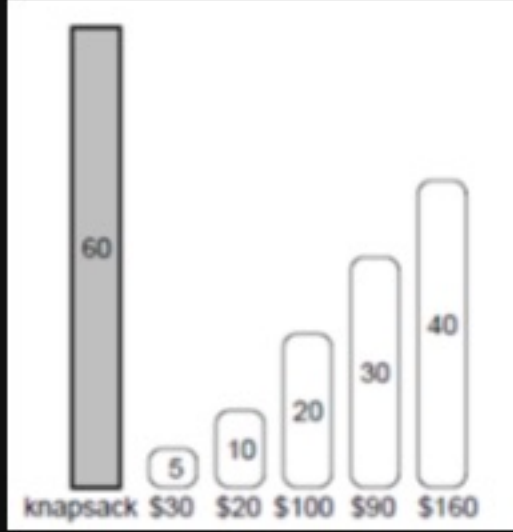
```
function g(n)
  if  $n \leq 1$  then return(n)
  else return( $5 \cdot g(n-1) - 6 \cdot g(n-2)$ )
```

Şekilde verilen algoritmanın  $n \geq 0$  için hesap yükü ifadesi nedir?

- A ☐  $n^{2n}$  yani  $\{n \times 2 \text{ Üssü } n\}$
- B ☐  $3n - 2n$  yani  $\{3 \text{ Üstü } n - 2 \text{ Üstü } n\}$
- C ☐  $3nn^3$  yani  $\{3 \text{ Üstü } n \times n^3\}$
- D ☐  $3n^{2n}$  yani  $\{3 \text{ Üssü } n \times 2 \text{ Üssü } n\}$
- E ☐  $n^{22n}$  yani  $\{n^{\text{Kare}} \times 2 \text{ Üstü } n\}$

Seçimi Boş Bırakmak İstiyorum





Şekilde bilgileri verilen Kesirli ve 0/1 Sırt çantası problemini Greedy algoritması ile çözülmektedir. Hangisi yanlıştır?

- A** ☐ Kesirli sırt çantası: 270 \$
- B** ☐ Greedy yaklaşımı 0/1 sırt çantası problemde optimal çözümü üretmektedir.
- C** ☐ 0/1 Sırt Çantası: 220 \$
- D** ☐ 0/1 Sırt Çantası optimal çözümü: 260 \$
- E** ☐ Greedy yaklaşımı kesirli sırt çantası problemde optimal çözümü üretmektedir.

Seçimi Boş Bırakmak İstiyorum

```
 $n = 2^{2^k}$  olsun.  
 $count \leftarrow 0$   
for  $i \leftarrow 1$  to  $n$   
   $j \leftarrow 2$   
  while  $j \leq n$   
     $j \leftarrow j^2$   
     $count \leftarrow count + 1$   
  end while  
end for  
return  $count$ 
```

Şekildeki algoritmada  $c=c+1$  deyiminin işletilme sayısı  $n$  eleman sayısına göre nedir?

- A ☐  $O(n^3)$
- B ☐  $O(\log n)$
- C ☐  $O(n^2)$
- D ☐  $O(n)$
- E ☐  $O(n \log \log n)$

Aşağıda verilen algoritmaların çalışma zamanı analizine ilişkin verilen büyüme hızı hangisinde yanlış olarak verilmiştir? ( $n^2$ ;Karesel)

- A** ☐ Quick Sort: Eniyi durum:  $O(n \log n)$  -En kötü durum:  $O(n^2)$
- B** ☐ Selection Sort : Eniyi durum:  $O(n^2)$  -En kötü durum:  $O(n^2)$
- C** ☐ Merge Sort: Eniyi durum:  $O(n \log n)$  -En kötü durum:  $O(n \log n)$
- D** ☐ Insertion Sort: Eniyi durum:  $O(n)$  -En kötü durum:  $O(n^2)$
- E** ☐ Heapsort : Eniyi durum:  $O(n \log n)$  -En kötü durum:  $O(n^2)$

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  |
| 1 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  |
| 2 | 0 | 0 | 3 | 4 | 4 | 7 | 7 | 7 | 7 | 7  |
| 3 | 0 | 0 | 3 | 4 | 4 | 7 | 8 | 9 | 9 | 12 |
| 4 | 0 | 0 |   |   |   |   |   |   |   |    |

Sırt Çantası (0/1) Problemi Dinamik programlama yaklaşımı ile çözülürse, tablonun boşluk kısmına hangi değerler gelmelidir?  $W=9$ kg, Ağırlıklar, 2, 3, 4, ve 5kg ve değerleri sırası ile 3, 4, 5 ve 7 TL'dir.

- A** ☐ 3, 4, 5, 7, 8, 10, 11, 12
- B** ☐ 3, 4, 4, 7, 10, 10, 11, 12
- C** ☐ 3, 4, 4, 7, 8, 9, 10, 12
- D** ☐ 3, 4, 4, 7, 8, 9, 11, 12
- E** ☐ 3, 4, 4, 7, 8, 9, 9, 12

```
function korona( $n$ )  
   $r := 0$ ;  
  for  $i := 1$  to  $n - 1$  do  
    for  $j := i + 1$  to  $n$  do  
      for  $k := 1$  to  $j$  do  
         $r := r + 1$   
  return( $r$ )
```

Aşağıdaki algoritmanın en kötü durum çalışma zamanı maliyeti Big (O) notasyonuna göre nedir?

- A ☐  $n(n+8)(n+1)(3n+4)$
- B ☐  $n(n-1)(2n-1)$
- C ☐  $(n-1)n(n+1)/2$
- D ☐  $n(n-1)/4$
- E ☐  $n(n^*n-1)/3$

Karmaşıklık sınıfları ile ilgili aşağıdaki ifadelerden hangisi yanlıştır?

- A** ☐ "Bir çözümü kontrol etmek her zaman kolay ise, bir çözüm bulmak kolay mı olmalı?" sorusu  $P = NP?$  olarak isimlendirilir ve çözümü bilinmeyen bir problemdir.
- B** ☐ Bir problemi tanımlayan non deterministik bir Turing makinesi varsa bu çözülemez ve izlenemez bir problemdir.
- C** ☐  $P$  sınıfında olmayan problemler çözülemez (unsolvable) problemlerdir.
- D** ☐ Herhangi bir NP-Complete problem polinomsal zamanda çözülebiliyorsa  $P = NP$  olur.
- E** ☐  $P = NP?$  Bir probleme yönelik bir çözümün doğru olup olmadığını kontrol etmek kolaysa, problemi çözmek de kolay mıdır? Anlamına gelmektedir.

Bir A dizisini sıralamak için, “öncelikle bu diziyi ikili arama ağacına yerleştirmek ve bu ağacı INORDER dolaşmak” biçiminde bir yol izlenmiştir. Bu düşünce biçimi hangi algoritma tasarım tekniği olarak değerlendirilmelidir?

- A** ☐ Azalt-Yönet algoritma tasarım tekniği
- B** ☐ Dönüştür-Yönet algoritma tasarım tekniği
- C** ☐ Aç Gözlü yaklaşım algoritma tasarım tekniği
- D** ☐ Kaba Kuvvet tasarım tekniği
- E** ☐ Böl-Yönet algoritma tasarım tekniği

Seçimi Boş Bırakmak İstiyorum

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ 2C(n/2) + bn \log n & \text{if } n \geq 2 \end{cases}$$

Aşağıda rekürsif tanımı verilen çalışma zamanının asimptotik gösterimi nedir?

- A** ☐  $O(\log n)$
- B** ☐  $O(n \log^2 n)$
- C** ☐  $O(n^2)$
- D** ☐  $O(n^3)$
- E** ☐  $O(n)$



$$f_1(n) = 10^n$$

$$f_2(n) = n^{1/3}$$

$$f_3(n) = n^n$$

$$f_4(n) = \log_2 n$$

$$f_5(n) = 2^{\sqrt{\log_2 n}}$$

Verilen algoritma büyüme hızların için aşağıdakilerden hangisi yanlıştır?

- A ☐  $f_2(n) = O(f_5(n))$
- B ☐  $f_1(n) = O(f_3(n))$
- C ☐  $f_4(n) = O(f_2(n))$
- D ☐  $f_4, f_2$  ile  $f_5$  arasındadır.
- E ☐  $f_5(n) = O(f_2(n))$

Left-to-Right binary exponentiation algoritması ile  $x^{60}$  ( $x$  üssü 60) değerinin hesaplanması ile ilgili olarak aşağıdakilerden hangisi doğrudur.

**A** ☐ Bu değer üs ten bağımsız olarak 1 sanlyede hesaplanır.

**B** ☐ Bu değer hesabi 6 adımda hesaplanablllr.

**C** ☐ Bu değer 3600 adımda hesaplanır

**D** ☐ Bu değer  $60 \times 5 = 300$  adımda hesaplanır

**E** ☐ Bu değer hesabi 59 adımda hesaplanablllr.

```
ALGORITHM UE ( $A[0..n-1]$ )  
  for  $i \leftarrow 0$  to  $n-2$  do  
    for  $j \leftarrow i+1$  to  $n-1$  do  
      if  $A[i] = A[j]$  return false  
  return true
```

Şekilde kaba kod olarak verilen algoritmanın en kötü durum davranışı ne zaman oluşur?

- A ☐ A dizisi tersten sıralı bir dizi ise
- B ☐ A dizisinin durumu algoritmanın çalışma zamanına etki etmez.
- C ☐ A dizisinin bütün elemanları eşit ise
- D ☐ A dizisi zaten sıralı bir dizi ise
- E ☐ A dizisinin bütün elemanları farklı ise yada sadece son iki elemanın bir birine eşit ise

|   | M1:8X3; | M2:3x2; | M3:2X19; | M4:19X18; | M5:18X7 |
|---|---------|---------|----------|-----------|---------|
|   | 1       | 2       | 3        | 4         | 5       |
| 1 | 0       | 48      | 352      | 1020      |         |
| 2 |         | 0       | 114      | 792       | 978     |
| 3 |         |         | 0        | 684       | 936     |
| 4 |         |         |          | 0         | 2394    |
| 5 |         |         |          |           | 0       |

Aşağıdaki matris zincirinin minimum çarpma sayısı ile sonuçlanması için uygun parantezleme sırası dinamik programlama yaklaşımı ile bulunmak istenmiştir. Tablonun boş kısmı ne olmalıdır?

- A** ☐ 1096
- B** ☐ 1184
- C** ☐ 1136
- D** ☐ 1024
- E** ☐ 1120

Aşağıda rekürsif tanımı verilen algoritmanın çalışma zamanının asimptotik ifadesi hangisidir?

$$T(n) = 2T(n/2) + n \log n$$

- A** ☐  $O(n^2 \log n)$  {2 üs olarak değerlendirilsin., yani n kare}
- B** ☐  $O(n \log n)$
- C** ☐  $O(n^2)$  { karesel}
- D** ☐  $O(n)$
- E** ☐  $O(n \log^2 n)$  {2 üs olarak değerlendirilsin}

$$T(n) = aT(n/b) + cn^k$$

$a = 2, b = 2, \text{ ve } k = 1$

Şekilde rekürsif tanımı verilen algoritmanın çalışma zamanının asimptotik ifadesi hangisidir?

- A ☐  $O(\log n)$
- B ☐  $O(n \log n)$
- C ☐  $O(n^2)$
- D ☐  $O(n^2 \log n)$
- E ☐  $O(n)$

$$f(a) = 20, f(b) = 7, f(c) = 10, f(d) = 4 \text{ ve } f(e) = 18$$

Bir dosyada kullanılan karakterlerin kullanım sıklığı aşağıda verilmiştir. Sabit uzunluklu kodlamada her bir karakterin 3 bit uzunlukta olduğunu varsayarak, greedy yaklaşımını kullanarak değişken uzunluklu bir kodlama yapılması durumunda aşağıdakilerden hangisi yanlış olur?

- A** ☐ Yaklaşımın maliyeti  $O(n \log n)$  olarak elde edilmiştir.
- B** ☐ Yaklaşık% 27'lik bir tasarruf sağlanmaktadır.
- C** ☐ Buradaki Greedy yaklaşımın bellek maliyeti  $O(n)$ 'dir.
- D** ☐ Dosya sabit uzunluklu kodlamada 177 bit olur
- E** ☐ Sıkıştırılmış dosyanın bellek hacmi 129 bit olur.

```
 $c \leftarrow 0$   
for  $i \leftarrow 1$  to  $n$   
   $m \leftarrow \lfloor n/i \rfloor$   
  for  $j \leftarrow 1$  to  $m$   
     $c \leftarrow c + 1$   
  end for  
end for
```

Şekildeki algorithmada  $c=c+1$  deyiminin işletilme sayısı  $n$  eleman sayısına göre nedir?

- A ☐  $O(1)$
- B ☐  $O(\log n)$
- C ☐  $O(n^2)$
- D ☐  $O(n)$
- E ☐  $O(n \log n)$



Karmaşıklık sınıfları ile ilgili aşağıdakilerden hangisi yanlıştır?

- A** ☐ NP: Polinom zamanında doğrulanabilen tüm dillerin sınıfı olarak tanımlanır.
- B** ☐ Eğer bir problem NP ise VEYA NP-zor ise bu problem NP-tam sınıfındadır.
- C** ☐ NP-hard – En az NP kadar zordur. (NP’de olmak zorunda değil);
- D** ☐ P: Polinom zamanında çözülebilen tüm karar problemlerinin sınıfıdır.
- E** ☐ NP-complete – NP’de çözülmesi en zor problemlerdir.

Seçimi Boş Bırakmak İstiyorum

Floyd Warshall Algoritması ile ilgili olarak aşağıdakilerden hangisi yanlıştır?

- A** ☐ Dinamik programlama algoritma tasarım tekniğini kullanır.
- B** ☐ Algoritmanın çalışma zaman maliyeti  $O(n^3)$  olarak gerçekleşir.
- C** ☐ Böl ve Yönet algoritma tasarım tekniğini kullanır.
- D** ☐ Algoritmanın çalışma zaman maliyeti  $O(n^2)$  olarak gerçekleşir.

Seçimi Boş Bırakmak İstiyorum

| A                      | B         | C         |
|------------------------|-----------|-----------|
| 300<br>TL              | 190<br>TL | 180<br>TL |
| 3 kg                   | 2 kg      | 2 kg      |
| Çanta Kapasitesi= 4 kg |           |           |

Sırt çantası kapasitesi 4 kg'dır. Çantaya alınabilecek ürünlerin ağırlıkları ve fiyatları şekilde gösterilmiştir. Kesirli sırt çantası için Greedy yaklaşımını uygulanırsa aşağıdaki çıkarımlardan hangisi yanlış olur?

- A ☐ 0/1 Sırt çantası olması durumunda çantanın değeri 370 TL'dir
- B ☐ 0/1 Sırt çantası olması durumunda Çantada B ve C ürünleri bulunur
- C ☐ Greedy yaklaşımının ürettığı sonuç optimum değildir
- D ☐ Çantanın değeri 395 TL olur
- E ☐ Çantada A'nın tamamı ve B'nin yarısı bulunur

B-Tree ağacı ve AVL ağacı için aşağıdakilerden hangisi yanlıştır?

- A** ☐ 2-3 ağaç için eniyi ve en kötü durum sırasıyla;  $\log_2(n+1)-1$  ve  $\log_3(n+1)-1$ . ( $\log_2$ , 2 tabanda ve  $\log_3$ , 3 tabanda)
- B** ☐ B tree ve AVL ağaçları dönüştür yönet tasarım tekniğinin bir parçası olurlar.
- C** ☐ n-düğümlü bir AVL ağacının maksimum yüksekliği  $\log_2(n)$  olur. ( $\log_2$ , 2 tabanda demektir)
- D** ☐ İkili arama ağacında eleman arama işleminin en kötü durumdaki maliyeti  $O(n)$  olarak hesaplanır.
- E** ☐ B Tree, Ağacı dengeli tutarken düğümün dallanma faktörünü de artırarak ağacın yüksekliğinin küçük olmasını hedefler.

Seçimi Boş Bırakmak İstiyorum

Aşağıdaki Problem ve Karmaşıklık sınıfı eşleştirmelerinden hangisi yanlıştır?

- A** ☐ Minimum Spanning Tree NP sınıfındadır.
- B** ☐ Hamiltonian Döngüsü NP-Complete {NP-tam} sınıfındadır.
- C** ☐ Quick sort algoritması P sınıfındadır
- D** ☐ Knapsack Problemi, NP-Complete {NP-tam} sınıfındadır.
- E** ☐ Gezgin satıcı problemi NP-Hard sınıfındadır.

Seçimi Boş Bırakmak İstiyorum

$$M_1 : 2 \times 3, \quad M_2 : 3 \times 6, \quad M_3 : 6 \times 4, \quad M_4 : 4 \times 2, \quad M_5 : 2 \times 7$$

|   | 1       | 2       | 3       | 4  | 5 |
|---|---------|---------|---------|----|---|
| 5 | 124/k=4 | 126/k=4 | 132/k=4 | 56 | 0 |
| 4 | 96/k=1  | 84/k=2  | 84      | 0  |   |
| 3 | 84 /k=2 | 72      | 0       |    |   |
| 2 | 36      | 0       |         |    |   |
| 1 | 0       |         |         |    |   |

Aşağıda verilen matrisi zincirinin çarpımında minimum sayıda işlem yapılması için matrislerin çarpım düzenini belirlemek için dinamik programlama yaklaşımı uygulanmış ve şekildeki sonuç elde edilmiştir. Bu sonuca göre matrisler hangi sırada çarpılmalıdır?

- A** ☐  $((M_1) \times (M_2 \times (M_3 \times M_4))) \times M_5$
- B** ☐  $((M_1 \times (M_2 \times M_3)) \times (M_4 \times M_5))$
- C** ☐  $((((M_1 \times M_2) \times M_3) \times (M_4 \times M_5)))$
- D** ☐  $((((M_1 \times M_2) \{M_3 \times M_4\}) \times M_5)$
- E** ☐  $((M_1 \times M_2) \{((M_3 \times M_4) \times M_5))$

Algoritma tasarım teknikleri ve örnekleri ile ilgili olarak aşağıdaki eşleştirmelerden hangisi yanlıştır?

- A ☐ Selection Sort –Brute Force
- B ☐ Insertion Sort–Decrease & Conquer
- C ☐ Prim Algorithm–Dynamic Programming
- D ☐ Merge Sort–Divide & Conquer
- E ☐ Heapsort–Transform & Conquer

Dinamik programlama paradigması ile Greedy paradigması karşılaştırıldığında aşağıdakilerden hangisi yanlıştır?

- A** ☐ Dinamik programlama Bottom-up tarzda çalışır
- B** ☐ Dinamik programlama optimum çözümü garantiler
- C** ☐ Greedy yaklaşımı Topdown çalışır
- D** ☐ Greedy yaklaşımı her zaman en iyi çözümü veremeyebilir
- E** ☐ Greedy yaklaşımının kurgusu daha karmaşık olarak gerçekleşmektedir.

Seçimi Boş Bırakmak İstiyorum



```
 $c \leftarrow 0$   
while  $n \geq 1$   
    for  $j \leftarrow 1$  to  $n$   
         $c \leftarrow c + 1$   
    end for  
     $n \leftarrow n/2$   
end while  
return  $c$ 
```

Aşağıdaki algoritmada  $c=c+1$  hesabının yapıldığı satırın hesap zaman maliyeti nedir?

- A ☐  $O(n \log n)$
- B ☐  $O(1)$
- C ☐  $O(n)$
- D ☐  $O(n^2)$
- E ☐  $O(\log n)$

$$M_1 : 5 \times 10, \quad M_2 : 10 \times 4, \quad M_3 : 4 \times 6, \quad M_4 : 6 \times 10, \quad M_5 : 10 \times 2$$

|              |                |                |                |                |
|--------------|----------------|----------------|----------------|----------------|
| $C[1,1] = 0$ | $C[1,2] = 200$ | $C[1,3] = 320$ | $C[1,4] = 620$ | ?              |
|              | $C[2,2] = 0$   | $C[2,3] = 240$ | $C[2,4] = 640$ | $C[2,5] = 248$ |
|              |                | $C[3,3] = 0$   | $C[3,4] = 240$ | $C[3,5] = 168$ |
|              |                |                | $C[4,4] = 0$   | $C[4,5] = 120$ |
|              |                |                |                | $C[5,5] = 0$   |

Zincir Matris Çarpımı Dinamik programlama algoritması aşağıdaki matris zincirine uygulandığında tablonun boşluk kısmına hangi değer gelmelidir?

A ☐ 326

B ☐ 462

C ☐ 548

D ☐ 348

E ☐ 388

|    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|---|---|---|---|---|---|---|---|---|---|----|----|----|
| 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  |
| 1  | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  |
| 2  | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| 3  | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3  | 3  | 3  |
| 4  | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4 | 4  | 4  | 4  |
| 5  | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 4 | 4 | 4  | 4  | 5  |
| 6  | 0 | 1 | 2 | 2 | 2 | 2 | 3 | 4 | 4 | 4 | 5  | 5  | 5  |
| 7  | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 5 | 5  | 5  | 5  |
| 8  | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5  | 5  | 6  |
| 9  | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 5 | 5 | 6  | 6  | 6  |
| 10 | 0 |   |   |   |   |   |   |   |   |   |    |    |    |

LCS algoritmasının dinamik programlama yaklaşımıyla A=XYXXZXVZY ve ZXZYVZXVXXZ için uygulanması halinde tablonun son satırı ne olmalıdır?

- A ☐ 1 2, 3, 4, 4, 4, 5, 5, 5, 6, 7, 7
- B ☐ 1 2, 3, 4, 4, 4, 5, 5, 6, 6, 7, 7
- C ☐ 1 2, 3, 4, 4, 4, 5, 6, 6, 6, 7, 7
- D ☐ 1 2, 3, 4, 4, 4, 5, 5, 6, 6, 6, 6
- E ☐ 1 2, 3, 4, 4, 4, 5, 5, 6, 6, 6, 7