

**Sivas Cumhuriyet Üniversitesi Mühendislik
Fakültesi Bilgisayar Mühendisliği Bölümü 2024-
2025 Eğitim Öğretim Yılı Bahar Algoritma
Analizi ve Tasarımı Course AAD321 Quiz 5**

1. Brute-Force algoritmalarının temel avantajı nedir?

- A) Her zaman en hızlı sonucu verir
- B) Bellek kullanımını en aza indirir

C) Tüm olası çözümleri deneyerek kesin çözüm sunar

D) Karmaşık veri yapılarını gerektirir

2. Aşağıdakilerden hangisi BruteForce algoritma -sına örnek değildir?

- A) Selection Sort
- B) Bubble Sort

C) Binary Search

D) Exhaustive Search

3. Selection Sort algoritmasının karmaşıklığı nedir?

- A) $O(n)$
- B) $O(\log n)$
- C) $O(n \log n)$

D) $O(n^2)$

4. Insertion Sort algoritması hangi stratejiye örnektir?

- A) Divide-and-Conquer
- B) Transform-and-Conquer

C) Decrease-and-Conquer

D) Dynamic Programming

5. Bubble Sort algoritmasında, her döngüde hangi özellik sağlanır?

A) En küçük eleman sona gelir

B) En büyük eleman sona itilir

C) Dizi ikiye bölünür

D) Ortadaki eleman seçilir

6. Aşağıdaki problemlerden hangisi exhaustive search yaklaşımıyla çözülür?

A) Binary Search

B) Merge Sort

C) Traveling Salesman Problem

D) Topological Sort

7. “Sahte Para” problemi aşağıdaki algoritma stratejilerinden hangisiyle çözülür?

A) Brute-Force

B) Decrease-by-a-Constant-Factor

C) Dynamic Programming

D) Greedy

8. Topolojik sıralama hangi veri yapısına uygulanır?

A) Yığın (Stack)

B) Ağaç (Tree)

C) Yönlü Aykırı Graf (DAG)

D) Döngüsel Liste

9. Aşağıdakilerden hangisi Knapsack probleminin çözümünde kullanılabilir?

A) Binary Search

B) Exhaustive Search

C) Quick Sort

D) Heap Sort

10. Permütasyon üretiminde kullanılan Johnson-Trotter algoritmasının özelliği nedir?

A) Her permütasyon ikili arama kullanır

B) Her yeni permütasyon yalnızca iki elemanın yer değiştirmesiyle oluşur

C) En küçük öğeyi başa yerleştirir

D) En büyük öğeyi sona iter

Her soru 2 puanlık çoktan seçmeli sorulardan oluşmaktadır.

11. Belirli bir üniversitede, öğrenciler kayıt ofisine teker teker gelir ve memur, her öğrencinin kayıt numarasını bir listeye ekler. Bu liste her an artan sırada kalacak şekilde düzenlenir. Her numara, listenin tamamı tekrar sıralanmadan doğrudan doğru konumuna yerleştirilir. Bu senaryo için en uygun algoritma türü nedir? Neden? Decrease and Conquer yöntemi kullanarak bir sayı listesini sıralayan algoritmayı yazınız. Yazdığınız bu algoritma için zaman verimliliği fonksiyonunu ($t(n)$) hesaplayınız, bu fonksiyon için büyük O kümesini bulunuz (25 Puan)

```
InsertionSort(A[0, ..., n-1]):
// Girdi : A uzunluğunda bir A dizisi
// Çıktı : Elemanları küçükten büyüye
sıralanmış A
    for i = 1 to n-1:
        key ← A[i]
        j ← i - 1
        while j ≥ 0 and A[j] > key:
            A[j + 1] ← A[j]
            j ← j - 1
        A[j + 1] ← key
```

- Insertion Sort

- yeni gelen elemanı mevcut sıralı liste içinde doğru konuma yerleştirerek çalışır. Bu durum tam olarak kayıt memurunun, gelen öğrencinin numarasını doğru yere yerleştirme işlemini yansıtır.

$$t(n) = 1 + 2 + 3 + \dots + (n-1) = \frac{n(n-1)}{2} \Rightarrow O(n^2)$$

12. 72 ve 41 olmak üzere iki tam sayı veriliyor. Bu iki sayının çarpımını çarpma operatörünü kullanmadan, Rus Çiftçisi (Russian Peasant) yöntemiyle hesaplayınız. Önce tabloyu oluşturunuz, ardından algoritma için sözde kodu (pseudo kodu) yazınız. (16 Puan)

n	m	c
72	41	0
36	82	0
18	164	0
9	328	0
4	656	+328
2	1312	0
1	2624	0
-	2624+328=2952	

```
RusKoyulusuCarpm (m, n)
//Girdi: m ve n tam sayıları
//Çıktı: m*n
toplam ← 0
While m ≠ 1 :
    if m mod 2 = 0
        m ← m / 2
    else
        m ← (m - 1) / 2
        toplam ← toplam + n
        n ← 2 * n
return toplam + n
```

13. Sıralı bir tamsayı A dizisi veriliyor. Bu dizi artan sırada (küçükten büyüğe) sıralanmıştır. Yapman gereken: Dizi yerinde (in-place) olacak şekilde, yani ek bir dizi kullanmadan, tekrar eden elemanları silmek. Elemanların sıralaması ilk görüldüğü haliyle aynı kalmalı. Fonksiyon, tekrarlar kaldırıldıktan sonra dizide kalan benzersiz elemanların sayısını (k) döndürmeli.

İşlem tamamlandıktan sonra A dizisinin ilk k elemanı yalnızca benzersiz elemanları içermeli. k'den sonraki elemanların değeri önemli değil ve dikkate alınmayabilir.

Örnek 1:

Girdi: A = [1, 1, 2]

Beklenen Çıktı: k = 2, A = [1, 2, _]

Örnek 2:

Girdi:A= [0, 0, 1, 1, 1, 2, 2, 3, 3, 4]

Beklenen Çıktı:k = 5, A= [0, 1, 2, 3, 4, _, _, _, _, _]

Not: k'den sonraki elemanların ne olduğunun önemi yoktur ve kontrol edilmez.

Not 2: Azalt ve Yönet (Decrease and Conquer) ya da Kaba Kuvvet (Brute Force) yöntemleri dışında başka hiçbir yöntem kullanamazsınız.

(14 Puan)

```
RemoveDuplicates(A[0, ..., n-1]):  
    if n == 0:  
        return 0  
    k ← 1  
    for i = 1 to n-1:  
        if A[i] ≠ A[k - 1]:  
            A[k] ← A[i]  
            k ← k + 1  
    return k
```

14. Bir Armstrong sayısı, basamaklarının küplerinin toplamı kendisine eşit olan sayıdır (3 basamaklı sayılar için klasik tanım).

Örneğin: $153 \rightarrow 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

Not: Kaba Kuvvet (Brute Force) yöntemi dışında başka hiçbir yöntem kullanamazsınız.

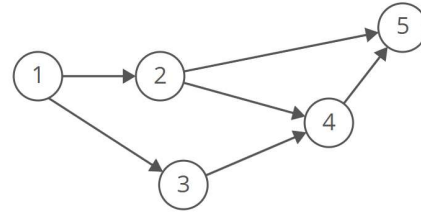
(12 Puan)

```
IsArmstrongNumber(n):  
    //Girdi:n Bir pozitif tamsayı. Bu sayı,  
    //Armstrong sayısı olup olmadığını kontrol  
    //etmek istediğiniz sayıdır.  
    //Çıktı:Eğer n bir Armstrong sayısı ise  
    //(yani, her basamağının küpünün toplamı  
    //kendisine eşitse), True döndürülür.  
    //Eğer n bir Armstrong sayısı değilse,  
    //False döndürülür.  
    original_number ← n  
    sum_of_cubes ← 0  
    while n > 0:
```

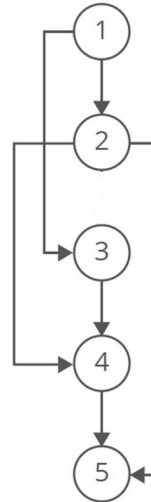
```
        digit ← n mod 10  
        sum_of_cubes ← sum_of_cubes +  
        digit^3  
        n ← n / 10 // Son basamağı çıkar  
    if sum_of_cubes == original_number:  
        return True  
    else:  
        return False // Sayı Armstrong  
sayısı değil
```

15. Şekilde birkaç düğüm (node) verilmiştir ve bunların topolojik sıralama (topological sort) kullanılarak sıralanması gerekmektedir.

DFS (Derinlik Öncelikli Arama - Depth First Search) algoritması kullanılarak bu düğümleri yeni düğümler olarak sıralayınız.



(13 Puan)



Süre 55dk Başarılar dilerim...

Eng: Abdulrahman Hamdi