

Cevap
Analizi

Ad-Soyadı: Abdulrahman Hamdi No:

Sivas Cumhuriyet Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
2024-2025 Eğitim Öğretim Yılı Bahar Algoritma Analizi ve Tasarımı Course AAD321 Vize Sınavı 3

1- Aldığı BubbleSort algoritması, uzunluğu n olan bir diziyi küçükten büyüğe sıralamak için kullanılmaktadır. Bu algoritmanın, en kötü durumda çalışırken gerçekleştirdiği karşılaştırma ve yer değiştirme işlemlerine bağlı olarak oluşan zaman karmaşıklığı $C_{worst}(n)$ nedir? Açıklayın

```
BubbleSort(A[0, ..., n-1])
// Girdi: n uzunluğunda bir A dizisi
// Çıktı: Elemanları küçükten büyüğe sıralanmış A dizisi

for i = 0'dan n-2'ye:           i = 0 olduğu anda içteki loop n-1 kez
    for j = 0'dan n-2-i'ye:       i = 1 " " " n-2 kez
        if A[j+1] < A[j]:         i = 2 " " " n-3 kez
            temp = A[j+1]         i = 3 " " " 0 kez
            A[j+1] = A[j]
            A[j] = temp           j = n-2 " " "
```

$$\Rightarrow C_{worst} = T(n) = (n-1) + (n-2) + \dots + 1 + 0 = \frac{(n-1)(n)}{2}$$

$$\Rightarrow C_{worst} = \frac{n^2 - n}{2} \in O(n^2)$$

2- Aşağıdaki problemlere belirtilen yöntemleri kullanarak algoritmalar tasarlayın ve sözde kodlarını yazın.

i) **Brute Force** yöntemi kullanarak bir dizide seçilen üç elemandan maksimum ile minimum değer arasındaki farkın, belirli bir eşik (Limit değeri) değerinden küçük veya eşit olup olmadığını kontrol eden bir algoritma yazınız.

ii) **Recursive** yöntemi kullanarak sıralı bir dizide, bir öğeyi bulmak için bir Binary Search (İkili Arama) algoritması yazın.

$$\log n = \frac{\ln n}{\ln 10}$$

3- Aşağıdaki problemlere belirtilen çözünüz:

i) Eğer $t_1(n) \in O(g_1(n))$ ve $t_2(n) \in O(g_2(n))$ $t_1(n) + t_2(n) \in O(\max(g_1(n), g_2(n)))$ olduğunu ispatlayınız.

ii) $t(n) = 3n^2 - 4n + 5$ fonksiyonunun $O(n^2)$ olduğunu gösteriniz.

$$t(n) \leq C \cdot g(n) \Rightarrow 3n^2 - 4n + 5 \leq C \cdot n^2$$

$$\begin{aligned} 3n^2 &\leq 3n^2 & n \geq 3 \\ -4n &\leq 4n^2 \\ +5 &\leq n^2 \\ \hline 3n^2 - 4n + 5 &\leq 8n^2 \\ \Rightarrow C &= 8, n_0 = 1 \\ \text{o halde } t(n) &\in O(n^2) \end{aligned}$$

4- Fonksiyonlarının büyüme derecelerini karşılaştırınız.

- a- n^3 ve 2^n
- b- $\log n$ ve n

$$a) t_1(n) = n^3 \text{ ve } t_2(n) = 2^n$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{n^3}{2^n} = \frac{\infty}{\infty} \Rightarrow \lim_{n \rightarrow \infty} \frac{3n^2}{\ln 2 \cdot 2^n} \Rightarrow \frac{1}{\ln 2} \cdot \lim_{n \rightarrow \infty} \frac{6n}{\ln 2 \cdot 2^n} \Rightarrow \frac{6}{(\ln 2)^2} \cdot \lim_{n \rightarrow \infty} \frac{1}{\ln 2 \cdot 2^n} = \frac{1}{\infty} = 0$$

= 0 olup $t_2(n)$ fonksiyonu $t_1(n)$ 'e göre daha hızlı büyür

$$b) \lim_{n \rightarrow \infty} \frac{\log n}{n} = \lim_{n \rightarrow \infty} \frac{1/n}{1} = \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

a2 önce aynı açıklama

$t_1(n) \in O(g_1(n)) \quad n \geq 0, \quad c_1 > 0, \quad n \geq n_0 \Rightarrow t_1(n) \leq c_1 \cdot g_1(n)$
 $t_2(n) \in O(g_2(n)) \quad n \geq 0, \quad c_2 > 0, \quad n \geq n_1 \Rightarrow t_2(n) \leq c_2 \cdot g_2(n)$

taraf taraf toplanırsa:

$$t_1(n) + t_2(n) \leq c_1 \cdot g_1(n) + c_2 \cdot g_2(n) \leq c_1 \cdot \max(g_1(n), g_2(n)) + c_2 \cdot \max(g_1(n), g_2(n))$$

$$\Rightarrow t_1(n) + t_2(n) \leq (c_1 + c_2) \cdot \max(g_1(n), g_2(n))$$

n_0 veya n_1 'den büyük olabilir ve $c = c_1 + c_2$ alınırsa $\Rightarrow t_1(n) + t_2(n) \leq c \cdot \max(g_1(n), g_2(n))$
 5- Aşağıdaki fonksiyonların zaman karmaşıklıklarını Big O notasyonu ile ifade edin

T(n)	Big O	T(n)	Big O
$T(n) = 3n^2 + 5n + 1$	$O(n^2)$	$T(n) = 4^n + n^5$	$O(4^n)$
$T(n) = 2^n + n^3 + 10$	$O(2^n)$	$T(n) = n^2 \log n + n$	$O(n^2 \log n)$

2) a) $VC \text{ Elemanı } (A[0, \dots, n-1], \text{target})$

//girdi: n uzunluğunda A dizisi, target izi verilen max fark

//çıkış: eğer A içinden seçilen her hangi bir üçlü eleman varsa bu üçlünün en büyüğü ile en küçük sayı

arasındaki fark target eşitse, 0 zaman dönen
 indiler, yoksa -1

for i ← 0 to n-3

for j ← i+1 to n-2

for k ← j+1 to n-1

min ← min(A[i], A[j], A[k])

max ← max(A[i], A[j], A[k])

if max - min = target

return [i, j, k]

return -1

Süre 55dk Başarılar dilerim...

Eng: Abdulrahman Hamdi

b)

Binary search(A[0, ..., n-1], k)

//girdi: n uzunluğunda küçükten büyüğe sıralı bir A dizisi ve bir k değeri

çıkış: k'nin olduğu A'nın indisi

alt ← 0

ust ← n-1

if ust ≥ alt

ort ← $\lfloor (ust + alt) / 2 \rfloor$

if A[ort] = k

return ort

elseif A[ort] > k

return Binary search(A, alt, ort-1, k)

else

return Binary search(A, ort+1, ust, k)

return -1