

a- Design an algorithm to solve this problem.

Ahmet KURT

290201034

Haas

TARGET_TEMP (days, temperatures)

max-temp = float('-inf')

for i = 0 to length of the days

for j = i + 1 to length of the days

temp-list = TEMP (days, temperatures, days[i], days[j])

if temp-list > max-temp

max-temp = temp-list

t-start = days[i]

t-end = days[j]

return t-start, t-end

We will use print() function in .py file.

b- Analyze and define the asymptotically tight complexity by reason.

⇒ For complexity we can analyze the pseudocode line by line.

⇒ Lines which are run once is;

1, 2 and 10 constant time $\Theta(1)$

⇒ Line 3 takes n (length of the days) time, runs n times. $\Theta(n)$

⇒ From line 4 run $\sum_{j=i+1}^n j = \frac{n \cdot (n+1)}{2} = \frac{n(n+1)}{2}$ times. $\Theta(n^2)$

⇒ Underline from line 4 also run n times. Because we are still

in a loop which takes n time. So;

⇒ $\Theta(n^2) \cdot \Theta(n) = \Theta(n^3)$ complexity

c- temperature-analysis.py

Q2

Ahmet KURT

290201034

a - Write the recurrence relation.

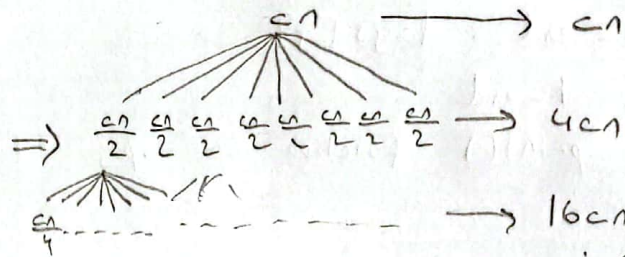
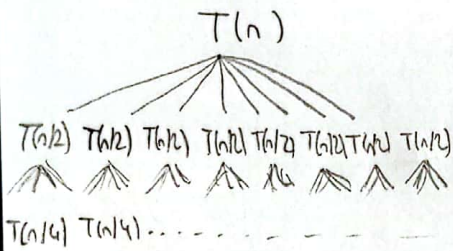
\Rightarrow We know that 8 subarrays each of which is of length $\lfloor n/2 \rfloor$

\Rightarrow We can show that like $T(n) = 8T(n/2)$,

Also question says the algorithm needs $\Theta(n)$ time to determine the subarrays where n is the number of elements in original array.

ANSWER: $T(n) = 8T(n/2) + n$

b - Use a recursion tree to determine asymptotic bound for the recurrence.



$$1 + x + x^2 + \dots + x^n = \frac{1 - x^{n+1}}{1 - x}$$

for $x \neq 1$

$$\Rightarrow cn \left(\frac{4^{1.5n+1} - 1}{4 - 1} \right) = \frac{4^{1.5n} \cdot 4 - 1}{3} = \frac{(4^{1.5n} - 1)cn}{3} = \frac{4cn^3 - cn}{3} = \Theta(n^3)$$

$$= cn(4^0 + 4^1 + 4^2 + 4^3 + \dots + 4^n)$$

$$1 + 4 + 4^2 + \dots + 4^n = \frac{1 - 4^{n+1}}{1 - 4}$$

ANSWER: $O(n^3)$

c - Use substitution method to verify your answer.

\Rightarrow We know the recurrence relation: $T(n) = 8T(n/2) + n$ and guess $\Theta(n^3) \rightarrow cn^3$

\Rightarrow Let's use substitution method;

$$\Rightarrow T(n) = 8T(n/2) + n \leq 8c\left(\frac{n}{2}\right)^3 + n = \underbrace{cn^3}_{\text{desired}} + \underbrace{n}_{\text{residual}}$$

! didn't work so try $cn^3 - dn$

$$\Rightarrow T(n) = 8T(n/2) + n \leq 8\left(c\left(\frac{n}{2}\right)^3 - \frac{dn}{2}\right) + n = cn^3 - 4dn + n$$

\Rightarrow BASE CASE

$$T(n) \leq cn^3 - dn$$

$$T(1) \leq c - d$$

c) d chosen.

$$= \underbrace{(cn^3 - dn)}_{\text{desired}} - \underbrace{(3dn - n)}_{\text{residual}}$$

$$3dn - n > 0$$

$$d > \frac{1}{3}$$

$$n > 0$$

as long as

$$d > \frac{1}{3}$$

$$n > 0$$