Ahmet KURT
290201034

a- Design an algorithm to solve this problem.

```
1  TARGET_TEMP (days, temperatures)
2      max_temp = float('-inf')
3      for i = 0 to legth of the days
4          for j = i+1 to legth of the days
5              temp_list = TEMP (days, temperatures, days [i], days [j])
6              if temp_list > max_temp
7                  max_temp = temp_list
8                  t_start = days [i]
9                  t_end = days [j]
10 return t_start, t_end
11 #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 $\sum_{j=i+1}^{n} j = \frac{n.(n+1)}{i+2} = \frac{n(n+1)}{2}$ times. $\Theta(n^2)$

=> Under from line 4 also run n times. Because we are still in a loop which takes n time. So;

=> $\Theta(n^2) . \Theta(n) = \Theta(n^3)$ complexity

---

c- temperature_analysis.py

Ahmet KURT
190201034

Q2

a - Write the recurrence relation.

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

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

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

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

---

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

$T(n)$

$T(n/2)\ T(n/2)\ T(n/2)\ T(n/2)\ T(n/2)\ T(n/2)\ T(n/2)\ T(n/2)$

$T(n/4)\ T(n/4) \ldots$

⇒ $cn \longrightarrow cn$

$\frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2}\ \frac{cn}{2} \longrightarrow 4cn$

$\frac{cn}{4} \ldots \longrightarrow 16cn$

$1 + x + x^2 + \ldots x^n = \frac{1 - x^{n+1}}{1 - x}$ for $x \neq 1$

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

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

$$\Rightarrow cn\left(\frac{4^{\lg n + 1} - 1}{4 - 1}\right) = \frac{4^{\lg n} \cdot 4 - 1}{3} = \frac{(4n^2 - 1)n}{3} = \frac{4cn^3 - cn}{3} = n^3$$

ANSWER: $O(n^3)$

---

c - Use substition method to verify your answer.

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

⇒ Let's use substition method;

⇒ $T(n) = 8\,T(n/2) + n \leq 8c\left(\frac{n}{2}\right)^3 + n = cn^3 + n$ → desired, → residual

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

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

$= (cn^3 - dn) - (3dn - n)$ → desired → residual

$3dn - n \geq 0$

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

$n \geq 0$

as long as $d \geq \frac{1}{3}$, $n \geq 0$

⇒ BASE CASE

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

$T(1) \leq c - d$

$c > d$ choosen.

CamScanner ile tarandı