Burak Yıldırım
1901042609

Q1) a) $T(n) = 16T(\frac{n}{4}) + n!$

$n^{\log_4 16} = n^2 < n! \rightarrow$ case 3

$16 \cdot (\frac{n}{4})! < c \cdot n! \rightarrow c$ can be found

$T(n) = \Theta(n!)$

b) $T(n) = \sqrt{2}\, T(n/4) + \log n$

$a = \sqrt{2} \quad b = 4 \quad f(n) = \log n$

$n^{\log_b a} = n^{\log_4 \sqrt{2}} = n^{1/4} > \log n \rightarrow$ case 1

$T(n) = \Theta(n^{\log_4 \sqrt{2}}) = \Theta(n^{1/4})$ //

c) $T(n) = 8T(\frac{n}{2}) + 4n^3$

$a = 8 \quad b = 2 \quad f(n) = 4n^3$

$n^{\log_2 8} = n^3 \quad 4n^3 = \Theta(n^3) \rightarrow$ case 2

$T(n) = \Theta(n^3 \log n)$ //

d) $T(n) = 64\, T(\frac{n}{8}) - n^2 \log n$

$a = 64 \quad b = 8 \quad f(n) = -n^2 \log n$

This can't be solved using master theorem because f(n) has to be asymptotic positive but here f(n) is negative.

e) $T(n) = 3T(\frac{n}{3}) + \sqrt{n}$

$a = 3 \quad b = 3 \quad F(n) = \sqrt{n} = n^{1/2}$

$n^{\log_3 3} = n > n^{1/2} \rightarrow$ case 1

$T(n) = \Theta(n^{\log_3 3}) = \Theta(n)$ //

f) $T(n) = 2^n T(\frac{n}{2}) - n^n$

This can't be solved using master theorem because f(n) has to be positive but here it is negative.

g) $T(n) = 3T(\frac{n}{3}) + \frac{n}{\log n}$

This can't be solved using master theorem because there is a non-polynomial difference between n and n/logn.

Q2) a) $X(n) = 9X(\frac{n}{3}) + n^2$

$a = 9 \quad b = 3 \quad f(n) = n^2$

$n^{\log_3 9} = n^2 \qquad n^2 = n^2 \to$ case 2

$X(n) = O(n^2 \log n)$

b) $Y(n) = 8Y(\frac{n}{2}) + n^3$

$a = 8 \quad b = 2 \quad f(n) = n^3$

$n^{\log_2 8} = n^3 \qquad n^3 = n^3 \to$ case 2

$Y(n) = O(n^3 \log n)$

c) $Z(n) = 2Z(\frac{n}{4}) + n^{1/2}$

$a = 2 \quad b = 4 \quad f(n) = n^{1/2}$

$n^{\log_4 2} = n^{1/2} \qquad n^{1/2} = n^{1/2} \to$ case 2

$Z(n) = O(\sqrt{n} \log n)$

I would choose algorithm 2 because since logn is common in all 3 of them when I compare the remaining parts $\sqrt{n}$ is smaller than other and this would provide smaller running time.

Q3) a) i. | 5 | 1 | 7 | 3 | 6 | 2 | 8 | 4 | → max comparisons (17)

In merge part, algorithm has to pick from different side for each iteration. This increases the amount of comparisons.

ii. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | → min comparisons (12)

In merge part, algorithm has to finish one side then the other side. This decreases the amount of comparisons.

b) i. | 4 | 5 | 6 | 8 | 7 | 1 | 3 | 2 | → max swap

Because every time a pivot is selected it's middle value of the remaining part, so all values are swapped.

ii. | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | → min swap (7)

Because every time a pivot is selected it's smallest value of the remaining part, so non of the values are swapped.

Q4) $T(n) = T(\frac{n}{2}) + 1$

$a = 1 \quad b = 2 \quad f(n) = 1$

$n^{\log_2 1} = 1 \qquad 1 = 1 \rightarrow$ case 2

$T(n) = \Theta(1 \cdot \log n) = \Theta(\log n)$

$\Rightarrow T(n) = O(\log n)$ because there is an if statement which can terminate the algorithm even if the running time is below $\log n$.