Fatih Doğaş

1)

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

$n^{\log_b a} = n^{\log_4 16} = n^2$ $\quad n! > n^2 \Rightarrow f(n) = \Omega(n^{\log_4 16} + \epsilon)$

$16\, f(n/4) <= c \cdot f(n) \Rightarrow 16\,(\frac{n}{4})! <= c \cdot n! \rightarrow$ $\boxed{\text{it is possible.}}$

$n>0$

So $\theta(f(n)) = \underline{\theta(n!)}$

case 3

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

$a = \sqrt{2}$

$b = 4$ $\quad f(n) = \log n$

$n^{\log_b a} = n^{\log_4 2^{\frac{1}{2}}} = n^{\frac{1}{4}} = \sqrt[4]{n}$ $\quad \log n > \sqrt[4]{n}$ $\quad f(n) = \Omega(n^{\log_4 \sqrt{2}} + \epsilon)$

$\sqrt{2}\, f(n/4) <= c \cdot f(n) \Rightarrow \sqrt{2} \cdot \log n/4 < c \cdot \log n \rightarrow$ $\boxed{\text{it is possible}}$

So $\theta(f(n)) = \underline{\theta(\log n)}$

c) $8\,T(n/2) + 4n^3 = T(n)$

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

$b = 2$

$n^{\log_b a} = n^{\log_2 8} = n^3$

So $\theta(n^{\log_2 8} \cdot \log n) = \underline{\theta(n^3 \cdot \log n)}$

d) $T(n) = 64\,T(n/8) - n^2 \log n$

can't be solved with master theorem because $f(n)$ must be asymptotic and positive. $f(n)$ <u>is not positive.</u>

e) $T(n) = 3\,T(n/3) + \sqrt{n}$ $\quad n^{\log_b a} = n^1 = n$ $\quad f(n) = O(n^{\log_3 3} - \epsilon)$

$a = 3$ $\quad f(n) = \sqrt{n}$

$b = 3$

So $\theta(n^{\log_b a}) = \underline{\theta(n)}$

Fatih Doğan

1-)

f) $T(n) = 2^n T(n/2) - n^n$

can't be solved with master theorem because $f(n) = -n^n$ must be positive.

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

can't be solved with master theorem because $\frac{n}{\log n}$ is not polynomial.

2)

a) $T(n) = 9T(n/3) + n^2$

$a = 9$ $\quad f(n) = n^2$ $\quad n^{\log_b a} = n^2$

$b = 3$ $\quad f(n) = \Theta(n^2) = \Theta(n^{\log_3 9})$

So $\Theta(n^{\log_b a} . \log n) = \Theta(n^2 . \log n)$

b) $T(n) = 8T(n/2) + f(n)$ $\quad f(n) = O(n^3)$

$a = 8$

$b = 2$ $\quad n^{\log_b a} = n^3$

So $\Theta(n^{\log_b a} . \log n) = \Theta(n^3 . \log n)$

c) $T(n) = 2T(n/4) + f(n)$ $\quad f(n) = O(\sqrt{n})$

$a = 2$

$b = 4$

$n^{\log_b a} = n^{\log_4 2} = n^{\frac{1}{2}} = \sqrt{n}$

So $\Theta(n^{\log_b a} . \log n) = \Theta(\sqrt{n} . \log n)$

I would choose c. Because it is the fastest among them. And in today's circumstances, memory is not a problem.

Fatih Doğan

3) a,i) {1, 5, 3, 7, 2, 6, 4, 8}

| 1 | 5 | 3 | 7 | 2 | 6 | 4 | 8 |

| 1 | 5 | | 3 | 7 | | 2 | 6 | | 4 | 8 |

| 1 | 3 | 5 | 7 | | 2 | 4 | 6 | 8 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

This way every number will be comparried with each other. And this is the way of maximum number of comparisons.

a.ii) {1, 2, 3, 4, 5, 6, 7, 8}

Already sorted arrays will require minimum number of comparisons

Fatih Dağdaş

4) The algorithm solves problems by dealing with only one half of the array.

Because of if-else block, subproblem number is one.
Because only if or else is working.

$a = 1$ $\qquad$ $f(n) = O(1)$ because assigning is $O(1)$

$b = 2 \rightarrow$ half the size

$T(n) = 1 \cdot T(n/2) + 1$

$n^{\log_b^a} = n^{\log_2^1} = n^0 = 1$ $\qquad$ $f(n) = \Theta(1)$

So $\Theta(n^{\log_b^a} \cdot \log n) = \Theta(n^0 \cdot \log n) = \Theta(1 \cdot \log n) = \underline{\Theta(\log n)}$