1) $T(n) = aT(n/b) + f(n)$ where $a >= 1$ and $b > 1$

Three cases;

* if $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
* if $f(n) = \Theta(n^{\log_b a})$, than $T(n) = \Theta(n^{\log_b a} \log n)$
* if $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$ (and, $af(\frac{n}{b}) \le cf(n)$ for some $c < 1$ for all n sufficiently large), then $T(n) = \Theta(f(n))$.

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

$a = 16$

$b = 4$

$f(n) = n!$

$n^{\log_b a} = n^2$

Since $f(n)$ is asymptotically larger than $n^{\log_b a}$, case 3 of the master theorem asks us check wheter $af(\frac{n}{b}) \le cf(n)$ some $c < 1$.

$f(n) = \Omega(n^{\log_b a})$, $T(n) = \Theta(n!)$

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

$a = \sqrt{2}$

$b = 4$

$f(n) = \log n$

$n^{\log_b a} = n^{\frac{1}{4}}$

$f(n)$ is asymptotically smaller than $n^{\log_b a}$

Case 1: $f(n) = O(n^{\log_b a}) = O(n^{\frac{1}{4}})$

$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\frac{1}{4}})$

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

$a=8$
$b=2$
$f(n)=4n^3$
$n^{\log_b a} = n^3$

$f(n)$ is asymptotically the same as $n^{\log_b a}$

Case 2; $f(n) = \Theta(n^3)$

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

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

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

The cases where $f(n)$ is negative can not be solved with the master theorem.

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

$a=3$
$b=3$
$f(n)=\sqrt{n}$
$n^{\log_b a} = n$

$f(n)$ is asymptotically smaller than $n^{\log_b a}$

Case 1: $f(n) = O(n^{\log_b a}) = O(n)$

$T(n) = \Theta(n^{\log_b a}) = \Theta(n)$

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

$f(n) = -n^n$

Does not apply ($f(n)$ is not positive)

9) $T(n) = 3T\left(\frac{n}{3}\right) + \frac{n}{\log n}$

$$\left.\begin{array}{l} a=3 \\ b=3 \\ f(n)=\frac{n}{\log n} \\ n^{\log_b a}=n \end{array}\right\}$$

$f(n)$ is smaller than $n^{\log_b a}$ but by less than a polynomial factor. Therefore, the master theorem makes no claim about the solution to the recurrence

---

2-a) $T(n) = 9T\left(\frac{n}{3}\right) + O(n^2)$

Using master theorem;

$T(n) = 9T\left(\frac{n}{3}\right) + n^2$

$$\left.\begin{array}{l} a=9 \\ b=3 \\ f(n)=n^2 \\ n^{\log_b a}=n^2 \end{array}\right\} \begin{array}{l} f(n)=\Theta(n^2) \\ T(n)=O(n^2 \cdot \log n) \end{array}$$

2-b) $T(n) = 8T\left(\frac{n}{2}\right) + O(n^3)$

Using master theorem

$T(n) = 8T\left(\frac{n}{2}\right) + n^3$

$$\left.\begin{array}{l} a=8 \\ b=2 \\ f(n)=n^3 \\ n^{\log_b a}=n^3 \end{array}\right\} \begin{array}{l} f(n)=\Theta(n^3) \\ T(n)=O(n^3 \cdot \log n) \end{array}$$

2-c) $T(n) = 2T\left(\frac{n}{4}\right) + O(\sqrt{n})$

using master theorem

$T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$

$$\left.\begin{array}{l} a=2 \\ b=4 \\ f(n)=\sqrt{n} \\ n^{\log_b a} = \sqrt{n} \end{array}\right\} \begin{array}{l} f(n)=\Theta(\sqrt{n}) \\ T(n)=O(\sqrt{n}.\log n) \end{array}$$

Compare x and y

$$\lim_{n\to\infty} \frac{n^2.\log n}{n^3\log n} = \frac{1}{n} = 0 \Rightarrow n^2\log n \in O(n^3.\log n)$$

Compare x and z

$$\lim_{n\to\infty} \frac{n^2.\log n}{\sqrt{n}.\log n} = n^{\frac{3}{2}} = \infty \Rightarrow \sqrt{n}.\log n \in O(n^2.\log n)$$

Compare y and z

$$\lim_{n\to\infty} \frac{n^3.\log n}{\sqrt{n}.\log n} = n^{\frac{5}{2}} = \infty \Rightarrow \sqrt{n}.\log n \in O(n^3.\log n)$$

sort by fast $z > x > y$

I would choose z because fastest algorithm is z.

3-a-1) For maximum comparison, each element of the array must be compared at least once.

myArray = [5,1,7,3,6,2,8,4]

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

5,1 7,3 6,2 8,4 each pair will be compared

1,5 3,7 2,6 4,8 Every pair of set is used comparison

1,3,5,7

2,4,6,8

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

3-a-2) The array should be sequential or close to the sequential so that the number of comparisons is minimum.

myArray = [1,2,3,4,5,6,7,8]

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

1,2 3,4 5,6 7,8 exchange is not performed

minimum comparisons

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

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

3-b-1)

myArray = [25, 27, 35, 40, 30, 15, 10, 20]   Swap 27 and 20

Pivot left   right

[25, 20, 35, 40, 30, 15, 10, 27]   Swap 35 and 10

left   right

[25, 20, 10, 40, 30, 15, 35, 27]   Swap 40 and 15

left   right

[25, 20, 10, 15, 30, 40, 35, 27]   Swap 25 and 15

Pivot   left

[15, 20, 10, 25, 30, 40, 35, 27]

Pivot   Pivot

Swap 20, 10 [15, 20, 10]   [30, 40, 35, 27] Swap 40 and 27

Swap 15, 10 [15, 10, 20]   [30, 27, 35, 40] Swap 30 and 27

[10, 15, 20]   [27, 30, 35, 40]

Sorted myArray = [10, 15, 20, 27, 30, 35, 40]

For maximum swapping in quick sort, the left and right of the piv should be adjusted so that the compared elements are constan: changed.

CamScanner ile tarandı

3.3-b-2) myArray = [1,2,3,4,5,6,7,8]

↑ Pivot

Since the given array is sorted, there is no element to move to the right of the pivot. In each divide operation, comparison will continue, but swap operation will not be applied.

3.

4) $T(n) = T(n/2) + 1 \quad$ if $n > 1 \quad T(1) = 1$

$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + 1$

$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + 1$

$T(n) = T\left(\frac{n}{2^2}\right) + 2c$

$= T\left(\frac{n}{2^3}\right) + 3c$

Recurrence Relation:

$T(n) = T\left(\frac{n}{2^k}\right) + 1 + 1 + 1 \ldots + 1$

$k = \log_2 n$

Since $T(1) = 1$, when $n = 2^k$, $T(n) = T(1) + k = 1 + \log_2 n$

$\log_2(n) \leq 1 + \log_2(n) \leq 2\log_2(n), \; \forall \; n \geq 2$

$T(n) = \Theta(\log_2(n))$

CamScanner ile tarandı