Samet Nalbant
180104 2614

1-

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

$a = 16$, $b = 4$, $f(n) = n!$

$f(n) = \Omega(n^{\log_b a + \epsilon})$, $a \geq 1$, $b > 1$

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

if $a \cdot f(n/b) \leq c \cdot f(n)$ for $n = 50$, $c = 2$

$16(n/4)! \leq c \cdot n!$ → $16 \cdot 12! \leq 2 \cdot 50!$ → $T(n) = \Theta(n!)$

b) $T(n) = \sqrt{2}T(\frac{n}{4}) + \log n$ $a = \sqrt{2}$, $b = 4$, $f(n) = \log n$ → $\Theta(n^{\log_4 \sqrt{2}})$ → $\Theta(n^{0.27})$

c) $T(n) = 8T(\frac{n}{2}) + 4n^3$ $a = 8$, $b = 2$ $f(n) = 4n^3$ $n^{\log_b a} = n^3$

$f(n) \neq \Theta(n^3)$ $T(n) = \Theta(n^3 \log n)$

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

$f(n) < 0$ and it can't be solved according to master theorem.

e) $T(n) = 3T(\frac{n}{3}) + \sqrt{n}$ $a = 3$, $b = 3$ $f(n) = \sqrt{n}$

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

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

It can't be solved according to master theorem. Because while algorithm is going further time can not decrease.

g) $T(n) = 3T(\frac{n}{3}) + \frac{n}{\log n}$ $a=3$, $b=3$ $f(n) = \frac{n}{\log n}$ $n^{\log_3 9} = n$

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

2-

a) $T(n) = 9T(\frac{n}{3}) + n^2$ $a=9$, $b=3$, $f(n) = n^2$

$n^{\log_b a} = n^2$, $T(n) = O(n^2 \log n)$ → according to master theorem.

b) $T(n) = 8T(\frac{n}{2}) + cn^3$ $a=8$, $b=2$, $f(n) = n^3$

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

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

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

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

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

To choose one of them, each expression must be compared.

x and z:

$$\lim_{n \to \infty} \frac{n^2 \log n}{\sqrt{n} \log n} = \frac{n^2}{n^{1/2}} = n^{\frac{3}{2}} \to \infty, \quad n^2 \log n > \sqrt{n} \log n$$

x z

z is faster than x

x and y:

$$\lim_{n \to \infty} \frac{n^2 \log n}{n^3 \log n} \to \frac{1}{n} = 0 \qquad n^2 \log n < n^3 \log n$$

x y

x is faster than y

* algorith z faster than x and algorithm x faster than y. So z is faster than x and y.

3)

a-1) Give an example of an 8 elaret array which requires the maximum number of comparesions.

* For the maximum number of comparisons every elaret in array must be compared in merging.

Ex: [9, 6, 5, 8, 7, 4, 3, 10]

9 6 5 8 7 4 3 10

96 58 74 310

69 58 47 310

5869 3470

= 3456789 10 =

a-2) Give an example of an 8 elaret array which requires the minimum number of comparesions.

* If the array is already sorted in increasing order number of comparisons minimum.

Ex: 3, 4, 5, 6, 7, 8, 9, 10

3456 7 8910

34 56 78 910 → there will be min. comp.

there will be min comp

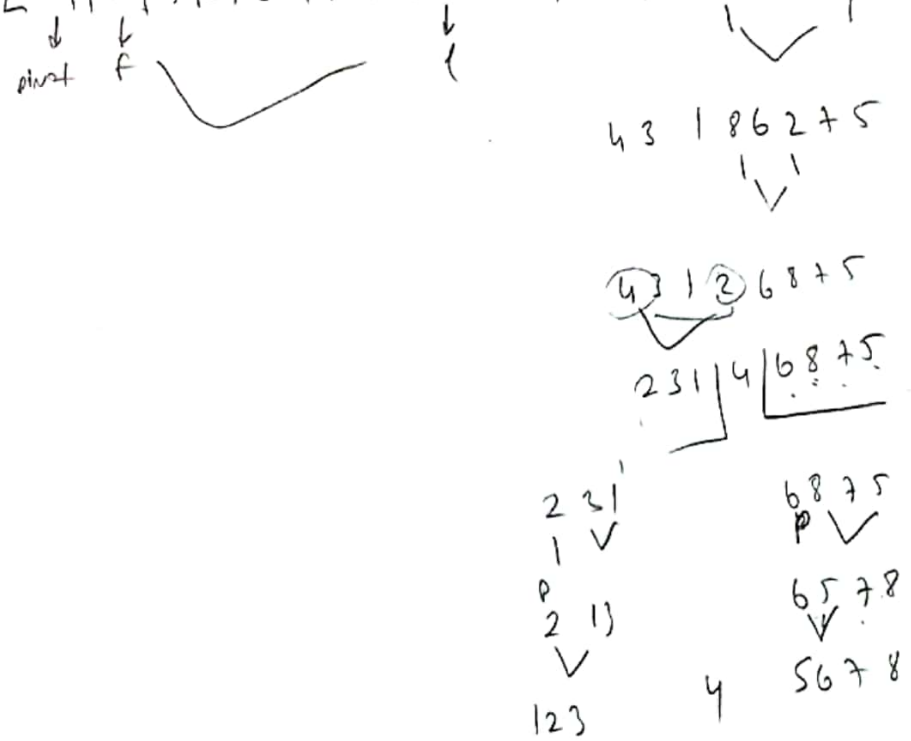
3456 7890

3-b consider the quick sort.

1) give an example of an 8 element array which requires the maximum number of swap operations. (Assume the pivot is the first)

for maximum swap operation pivot must be middle element and the lower elements than must be placed end of array, bigger elements must be placed in beginning of array.

Ex: [4, 5, 7, 8, 6, 2, 1, 3] + 4, 3, 7 8 6 21 5



2) give an example of an 8 element array which requires the minimum

for minimum swap operation if the array is already sorted there won't be any swap operations, however it's the worst case acording to time. Ex: 1, 2, 3, 4, 5, 6, 7, 8

4) Write a recurrence relation and find time complexity by deriving from the recurrence relation.

```
algorithm(left, right)                    → T(n)
    mid = (left + right) / 2              → O(1)
    if A[mid] == 0                        → O(1)
        return mid
    else
        if A[mid] > 0
            right = mid
            algorithm(left, right)        → T(n/2)
        else
            left = mid
            algorithm(left, right)        → T(n/2)
```

$T(n) = T(n/2) + 1$ $\quad a = 1 \quad b = 2 \quad f(n) = 1 \quad n^{\log_2 1} = 1$

equal case 2

$O(n) = n^{\log_b a} \log n \rightarrow O(\log n)$