Dilara Karakaş
171044010

# CSE 321 HW3

1) a) $T(n) = 27\,T(n/3) + n^2$

From master theorem;

$a(n/b) + f(n) \Rightarrow$ if $a > b^d$, $\Theta(n^{\log_b a})$

$27 > 3^2$ ✓ so $\Theta(n^{\log_3 27}) = \boxed{\Theta(n^3)}$

b) $T(n) = 3\,T(n/4) + n$

from master theorem; $3 < 4^1$ so $\Theta(n^{\log_b a}) \Rightarrow \boxed{\Theta(n^{\log_4 3})}$

$\Rightarrow \log_4 3 \cong 1{,}58$ Therefore; $\boxed{\Theta(n^{1{,}58})}$

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

from master theorem; $2 = 4^{1/2}$ Therefore, $\Theta(\sqrt{n} \cdot \log n)$

d) $T(n) = 2\,T(\sqrt{n}) + 1$

$A(x) = T(2^x)$ $(2^x = n)$
$A(x) = 2\,T(2^{x/2}) + 1$
$A(x) = 2\,A(x/2) + 1$ $\longrightarrow$ with master theorem

so $2 > 2^0$ $\Theta(x^{\log_2 2}) = \Theta(x)$ and

$n = 2^x$ so $x = \log n$

$\Rightarrow \boxed{\Theta(\log n)}$

e) $T(n) = 2\,T(n-2)$, $T(0) = 1$, $T(1) = 1$

$r^2 = 2$
$r^2 - 2 = 0$ char eq. $r_{1,2} = \mp\sqrt{2}$

$\alpha \cdot (+\sqrt{2})^n + \beta(-\sqrt{2})^n = 0$

$T(0) = 1 \Rightarrow \alpha + \beta = 1$
$T(1) = 1 \Rightarrow \sqrt{2}\alpha - \sqrt{2}\beta = 1$ (+)

$2\sqrt{2}\,\alpha = \sqrt{2} + 1$

$\alpha = \dfrac{\sqrt{2}+1}{2\sqrt{2}}$

$\beta = \dfrac{\sqrt{2}-1}{2\sqrt{2}}$

so $\boxed{T(n) = \Theta\left(\left(\frac{\sqrt{2}+1}{2\sqrt{2}}\right)(\sqrt{2})^n + \left(\frac{\sqrt{2}-1}{2\sqrt{2}}\right)(-\sqrt{2})^n\right)}$

f) $T(n) = 4T(n/2) + n$ , $T(1) = 1$

from master theorem

$4 > 2^1$ , so $\Theta(n^{\log_b a}) = \Theta(n^{\log_2 4}) = \boxed{\Theta(n^2)}$

g) $T(n) = 2T(\sqrt[3]{n}) + 1$ , $T(3) = 1$

$A(x) = T(2^x)$

$= 2T(2^{\frac{x}{3}}) + 1$

$A(x) = 2A(x/3) + 1$

from master theorem

$\log_3 2 > 0$ so $\Theta(k^{\log_3 2})$

$k = (\log n)$ so $\Theta(\log n^{\log_3 2}) =$

$= \boxed{\Theta(\log_3 2 \cdot \log n)}$

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

loop | recursive call | 1 print

$n > 2^0$ $2^k > 2^0$

$\hookrightarrow k$ must be $> 0$

Therefore $\boxed{\Theta(n^{\log n})}$

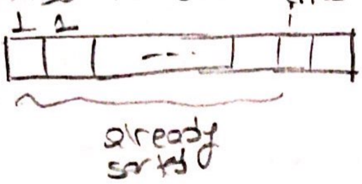
3) Algorithm first performs a constant-time $(+1)$ and then recursively calls 3 times. Recursive calls divides array $2/3$.

$T(n) = 3T(2/3\, n) + 1$

from master theorem; $3 > (3/2)^0$ so $\boxed{\Theta(n^{\log_{3/2} 2})}$

## 4) Insertion Sort Average Case;

Insertion sort tries to sort the input in a reserve order




So algorithm tries to put $i^{th}$ element from $i-1^{th}$ index

We need to sum all operations for every index.

$$T = T_1 + T_2 + \dots + T_n = \sum_{i=1}^{n-1} T_i$$

There is a 2 probobality when we compare 2 element. If $i^{th}$ index is smaler than $i-2^{th}$ index or not. Both of the case has 1/2 probability.

We have n this operation in raverge and we need to apply it for every element until array sorted.

Also for every element we have 1/2 probability to swap operation.

$$O(n)\, O(n/2) = O(n^2/2) = \boxed{O(n^2)}$$

## Quick sort Average sort;

The algorithm can choose any elemen as a pivot after first iteration

$$A(n) = \underbrace{\text{operation in rearrage}}_{\text{high-low+2}} + \text{recursive calls}$$

Pivot element can be placed any element so, every element has probability 1/n.

$$A(n) = (\text{high} + \text{low} - 2) + \sum_{x} [T_2 \mid \mathbb{X}x] \cdot 1/n \quad (x = \text{position of pivot})$$

CamScanner ile tarandı

$$A(n) = (n+1) + \sum_{i=1}^{n} \underbrace{[A(i-1) + A(n-i)]}_{} \cdot \frac{1}{n}$$ → probability

↳ this part represents left and right part of the pivot element.

$$A(n) = (n+1) \cdot 2/n \, [A(0) + A(1) + \dots + A(n-1)]$$

↳ $A(n)$ to $\frac{A(n)}{n+1} - \frac{A(n-1)}{n}$ we obtain $\frac{2}{n+1}$

We have;

$$\frac{A(n)}{n+1} - \frac{A(n-1)}{n} = \frac{2}{n+1}$$

$$T(n) = \frac{A(n)}{n+1} \Rightarrow T(n) = T(n-1) + \frac{2}{(n+1)}$$

→ $T(0) = 0$
Lets apply backward sub. for solving this recurrence relation

$$T(0) = \frac{A(0)}{1} = 0$$

$$T(n) = \sum_{i=2}^{n} \frac{2}{i+1} = 2 \underbrace{H(n+1)}_{\text{harmonic series}} - 3$$

Therefore $\Rightarrow A(n) = T(n) \cdot (n+1) = 2 \cdot (n+1) \cdot \underbrace{H(n+1)}_{\ln(n+1)} - 3(n-1) \in \boxed{\Theta(n \log n)}$

When I run the program for size of 1000 elements.

Quick sort = 6157
Insertion sort = 245106

As can be seen from these results, quick sort is much more efficient.

CamScanner ile tarandı

5) a) Dividing into 5 subproblems with one-third of the size then combining them in quadratic time

$$T(n) = 5\,T(n/3) + n^2$$

master theorem: $5 < 3^2$ so $\boxed{\Theta(n^2)}$

b) $T(n) = 2T(n/2) + n^2$

master theorem $\exists$ $2 < 2^2$ so $\boxed{\Theta(n^2)}$

c) $T(n) = T(n-1) + n$

$$T(n-1) = T(n-2) + n - 1$$
$$T(n-2) = T(n-3) + n - 3$$
$$\vdots$$
$$+ \quad T(1) = T(0) + 1$$

$$T(n) = T(0) + (1 + 2 + \dots + n)$$

$$T(n) = T(0) + \frac{n.(n+1)}{2} = \frac{n^2+n}{2}$$

$$\Theta\left(\frac{n^2+n}{5}\right) = \boxed{\Theta(n^2)}$$