Süleyman KORAMAZ
1901042615

CSE321 HOMEWORK-2

1. (a) $T(n) = 3T(n-1) - 2T(n-2)$

$\Rightarrow \alpha^2 = 3\alpha - 2 \to \alpha^2 - 3\alpha + 2 = 0 \to \alpha_1 = 2 \quad \alpha_2 = 1 \to T(n) = c_1 \cdot 2^n + c_2 \cdot 1^n$ $\quad \left.\begin{array}{l} T(1) = 2c_1 + c_2 = 1 \\ T(2) = 4c_1 + c_2 = 2 \end{array}\right\} \Rightarrow \begin{array}{l} c_1 = \frac{1}{2} \\ c_2 = 0 \end{array}$

$\Rightarrow T(n) = \frac{2^n}{2} + 0 \cdot 1^n = 2^{n-1} \longrightarrow T(n) = 2^{n-1} \to T(n) \in \Theta(2^n)$

(b) $T(n) = \underbrace{T(n/2)}+1 \quad T(\frac{n}{2}) = \underbrace{T(\frac{n}{2^2})}+1 \quad T(\frac{n}{2^2}) = T(\frac{n}{2^3})+1$

$\Rightarrow T(n) = T(\frac{n}{2}) + 1 = T(\frac{n}{2^2}) + 1 + 1 = T(\frac{n}{2^3}) + 1 + 1 + 1 = T(\frac{n}{2^k}) + \underbrace{1 + 1 + \dots + 1}_{k}$

$\Rightarrow T(n) = T(\frac{n}{2^k}) + k = 1 + k \to n = 2^k \to k = \log_2 n \xrightarrow{T(1)=1} T(n) = \log_2 n + 1 \Rightarrow T(n) \in \Theta(\log_2 n)$

(c) $T(n) = \underbrace{4T(n-1) - 4T(n-2)}_{T_H(n)} + \underbrace{3n}_{T_p(n)} \to T(n) = T_H(n) + T_p(n) \longrightarrow T_p(n) = An + B$

$T_H(n) \Rightarrow \alpha^2 = 4\alpha - 4$
$\alpha^2 - 4\alpha + 4 = 0 \quad \alpha_1 = 2, \ \alpha_2 = 2$
$\left.\right\} T_H(n) = c_1 \cdot 2^n + c_2 \cdot n \cdot 2^n$, $T_p(n) = 3n + 12$

$\begin{cases} T(n) = An + B \\ T(n-1) = A(n-1) + B = An - A + B \\ T(n-2) = A(n-2) + B = An - 2A + B \end{cases}$

$T(n) = An + B = 4 \cdot (An - A + B) - 4(An - 2A + B) + 3n$
$An + B = \cancel{4An} - 4A + \cancel{4B} - \cancel{4An} + 8A - \cancel{4B} + 3n$
$An + B = 4A + 3n \to A = 3, \ B = 12$

$T(n) = c_1 \cdot 2^n + c_2 \cdot n \cdot 2^n + 3n + 12 \to T(n) \in \Theta(n \cdot 2^n)$

c_2 is constant

(d) $T(n) = 4T(\frac{n}{2}) + n^2 \to$ Master Theorem $\to \begin{array}{l} a = 4 \\ b = 2 \end{array} \quad \begin{array}{l} f(n) = n^2 \\ d = 2 \end{array} \Big\} \begin{array}{l} a = b^d \\ 4 = 4 \end{array} \Big\} T(n) \in \Theta(n^d \cdot \log n)$ $(d = 2)$

$\Rightarrow T(n) \in \Theta(n^2 \cdot \log n)$

(e) $T(n) = 2T(\frac{n}{2}) + \Theta(n) \to$ Master Theorem $\to \begin{array}{l} a = 2 \\ b = 2 \end{array} \quad \begin{array}{l} f(n) = n \\ d = 1 \end{array} \Big] \begin{array}{l} a = b^d \\ 2 = 2 \end{array} \Big\} T(n) \in \Theta(n^d \cdot \log n)$ $(d = 1)$

$\Rightarrow T(n) \in \Theta(n \cdot \log n)$

(f) $T(n) = \underbrace{T(\frac{n}{2}) + T(\frac{n}{4})}_{H} + \underbrace{n}_{P} \to T(n) = T_H(n) + T_p(n)$

$T_H(n) \Rightarrow \alpha^2 = \alpha + 1 \Rightarrow \alpha^2 - \alpha - 1 = 0$

$\Rightarrow \alpha_{1,2} = \frac{-b \mp \sqrt{\Delta}}{2} = \frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}$

$T_H(n) = c_1 \cdot (\frac{1+\sqrt{5}}{2})^n + c_2 \cdot (\frac{1-\sqrt{5}}{2})^n$

$T_p(n) = An + B$
$T_p(\frac{n}{2}) = \frac{An}{2} + B$
$T_p(\frac{n}{4}) = \frac{An}{4} + B$

$An + B = \frac{An}{2} + B + \frac{An}{4} + B + n$
$An = \frac{3An}{4} + n \to n = \frac{1}{4}A \to A = 4n$
$B = 2B \to B = 0, \ T_p(n) = 4n + 0$

$\Rightarrow T(n) = c_1 \cdot (\frac{1+\sqrt{5}}{2})^n + c_2 \cdot (\frac{1-\sqrt{5}}{2})^n + 4n \to T(n) \in \Theta((\frac{1+\sqrt{5}}{2})^n)$

c_1 is constant

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

$T(2) = T(1) + 2$
$T(4) = T(2) + 4 = T(1) + 2 + 4$
$T(8) = T(1) + 2 + 4 + 8$
$\vdots$
$T(n) = \underbrace{T(1) + 2 + 4 + \dots + n}_{k+1 \text{ terms}} \quad n = 2^k$

$\Rightarrow 1 \cdot \frac{2^{k+1} - 1}{2 - 1} = 2 \cdot 2^k - 1 \rightsquigarrow 2n - 1$

$\boxed{\begin{array}{l} T(n) = 2n - 1 \\ T(n) \in \Theta(n) \end{array}}$

(h) $T(n) = 2T(n^{\frac{1}{2}}) + 1 \quad T(2) = 1, \ T(4) = 3$

$T(2) = 1$
$T(4) = 2T(2) + 1 = 3$
$T(16) = 2 \cdot (2T(2) + 1) + 1 \Rightarrow 2^2 \cdot T(2) + 2 + 1$
$T(256) = 2 \cdot (2 \cdot (2T(2) + 1) + 1) + 1 \Rightarrow 2^3 \cdot \underbrace{T(2)}_{1} + 4 + 2 + 1$
$\vdots$
$T(n) = \underbrace{2^k + 2^{k-1} + \dots + 4 + 2 + 1}_{k} \rightsquigarrow n = 2^{2^k}$
$\Rightarrow \log_2 n = 2^k \cdot \cancel{\log_2 2}$

$1 \cdot \frac{2^{k+1} - 1}{2 - 1} \to 2 \cdot 2^k - 1 = \boxed{2 \cdot \log_2 n - 1}$

$\boxed{T(n) = 2 \cdot \log_2 n - 1} \ \boxed{T(n) \in \Theta(\log n)}$

2) a- Is_Balanced (BST)

```
procedure Is_Balanced (node)
    if node is null
        return True
    end if
    left_side = Is_Balanced (node.left)          ~> T(n/2)
    right_side = Is_Balanced (node.right)        ~> T(n/2)
    height_balance = height_of_tree(node.left) - height_of_tree(node.right)   ~> Θ(n)
    return (left and right and (height_balance <= 1))
end
```

$T(n) = 2 \cdot T(\frac{n}{2}) + \frac{\Theta(n)}{n}$

Master Theorem
$a = 2$, $b = 2$, $f(n) = n$, $d = 1$ → $a = b^d$ → $T(n) \in \Theta(n^d \cdot \log n)$

$\underbrace{}_{}$ $T(n) \in \Theta(n \cdot \log n)$

b- height_of_tree (BST)

```
procedure height_of_tree(node)
    if node is null
        return 0
    end if
    left_height = height_of_tree(node.left)      ~> T(n/2)
    right_height = height_of_tree(node.right)    ~> T(n/2)
    return (max(left, right) + 1)                ~> O(1)
end
```

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

Master Theorem → $a = 2$, $b = 2$, $f(n) = 1$, $d = 0$ → $a > b^d$ → $T(n) \in \Theta(n^{\log_b a})$

$T(n) \in \Theta(n)$

3) $A(n) = 5 \cdot A(\frac{n}{2}) + O(n^3)$

master theorem → $a = 5$, $b = 2$, $f(n) = n^3$, $d = 3$ → $a < b^d$ → $\boxed{A(n) \in \Theta(n^3)}$

$B(n) = 2 \cdot B(n-2) + O(n)$

$B(n) = 2 \cdot (2 \cdot B(n-4) + O(n)) + O(n)$

$B(n) = 2(2(2 \cdot B(n-8) + O(n)) + O(n)) + O(n)$

$B(n) = \underbrace{2^{\frac{n}{2}} \cdot T(0)}_{\text{constant}} + \underbrace{2^{\frac{n}{2}-1} \cdot O(n) + 2^{\frac{n}{2}-2} \cdot O(n) + \dots + O(n)}$

$\boxed{B(n) \in \Theta(2^{\frac{n}{2}} \cdot n)}$

$C(n) = 3 \cdot C(\frac{n}{2}) + O(n^2)$

master theorem → $a = 3$, $b = 2$, $f(n) = n^2$, $d = 2$ → $a < b^d$ → $\boxed{C(n) \in \Theta(n^2)}$

$\Theta(n^2) \subset \Theta(n^3)$ → $\Theta(n^2)$ is better → ? $\underbrace{n^2 < 2^{\frac{n}{2}} n \text{ or } 2^{\frac{n}{2}} n < n}$

$\Rightarrow \lim_{n \to \infty} \frac{n^2}{2^{\frac{n}{2}} \cdot n} = \lim_{n \to \infty} \frac{n}{2^{\frac{n}{2}}} \overset{\text{L'Hospital}}{=} \lim_{n \to \infty} \frac{1}{\frac{2^{\frac{n}{2}} \cdot \ln 2}{2}} = 0$

$\Rightarrow \Theta(n^2)$ is better than $\Theta(2^{\frac{n}{2}} n)$

$\boxed{\Rightarrow \text{I chose C}}$

4) Hopcroft Karp Algorithm:

1. Initialize M as empty
2. while there exists an P
   - Remove matching edges of P from M
   - Add not-matching edges of P to M
3. Return M

M → Maximum matching (matching of maximum number of edges)

P → A path in which the edges belong alternatively to the matching and not matching. P is also starts from and ends on free vertices.

$\boxed{E \to \text{edges}, V \to \text{vertices}}$

Best Case: Initial matching is already a maximum. In this case, program terminates after first BFS. BFS iteration takes $O(E)$ time → $O(E)$

Worst Case: In worst case, algorithm may need to perform a maximum number of iterations to find the P, which is $\sqrt{V}$. → $O(E \cdot \sqrt{V})$

Average Case: It is often influenced by the distribution of edges and vertices (V, E) and it takes $O(E \cdot \sqrt{V})$ time → $O(E \cdot \sqrt{V})$

5) foo(n):

```
    if n<=1:
        return 1
    else
        for i in range(n):
            print("x")
    return foo(n/2) + foo(n/2)
```

$O(n)$; $T(\frac{n}{2})$; $T(\frac{n}{2})$

$T(n) = 2 \cdot T(\frac{n}{2}) + O(n)$

master theorem
$a = 2$, $b = 2$, $f(n) = n$, $d = 1$ → $a = b^d$

$T(n) \in \Theta(n \cdot \log n)$