Ahmet Yusa Telli
151044092

1) a) $T(n) = 27T(\frac{n}{3}) + n^2$

$a = 27 \quad b = 3 \quad c = 2$ $\qquad 2 < \log_3 27$

$2 : \underbrace{\log_3 27}_{\log_3 3^3 = 3} \rightarrow 2 < 3$ $\qquad T(n) = \Theta(n^{\log_3 27}) = \Theta(n^3)$

b) $T(n) = 9T(n/4) + n$

$a = 9 \quad b = 4 \quad c = 1$

$1 < \log_4 9 \rightarrow T(n) = \Theta(n^{\log_4 9})$

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

$a = 2 \quad b = 4 \quad c = \frac{1}{2}$

$\frac{1}{2} = \underbrace{\log_4 2}_{\log_4 4^{\frac{1}{2}} = \frac{1}{2}} \Rightarrow \frac{1}{2} = \frac{1}{2}$ $\qquad T(n) = \Theta(\sqrt{n} . \log n)$

d) $T(n) = 2T(n/2) + 17$

$a = 2 \quad b = 2 \quad c = 0$

$0 < \underbrace{\log_2 2}_{1}$ $\qquad T(n) = \Theta(n^{\log_2 2}) = \Theta(n)$

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

Let $m = \log n \quad n = 2^m$ $\qquad T(2^m) = 2 . T(2^{m/2}) + 1$

$S(m) = T(2^m) = 4S(\frac{m}{2}) + 2$

$a = 4 \quad b = 2 \quad c = 0$ $\qquad 0 < 2 \Rightarrow S(m) = \Theta(m^2)$

$0 = \log_2 4 = \log_2 2^2 = 2$ $\qquad T(2^m) = \Theta(m^2)$

$T(n) = \Theta(\log^2 n)$

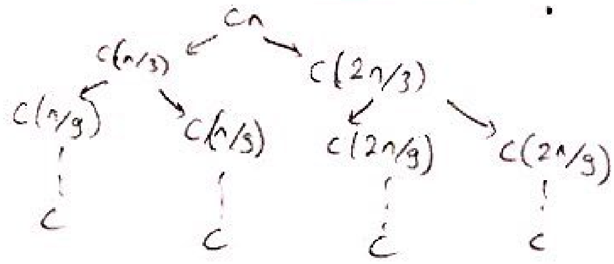
f) $T(n) = 4T(n/2) + n$

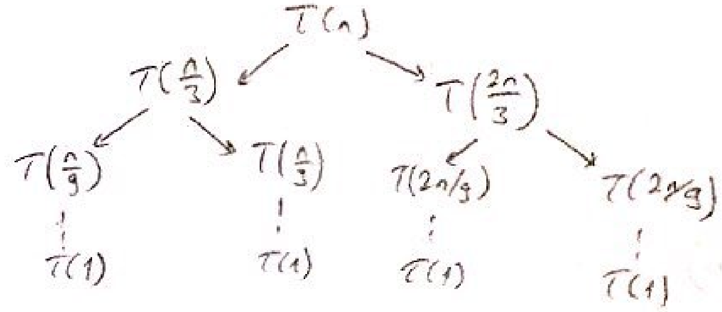
$a = 4 \quad b = 2 \quad c = 1$

$1 : \underbrace{\log_2 4}_{\underbrace{\log_2 2^2}_{2}}$ $\qquad 1 < 2$ $\qquad T(n) = \Theta(n^{\log_2 4}) = \Theta(n^2)$

g) $T(n) = \underbrace{T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)}_{\text{Looks like a Tree } T(n)} + \underbrace{O(n)}_{cn}$

If we think This is a $T(n)$ tree.





$T(n) = cn.\log_3 n = \theta(cn.\log_3 n)$

h) $T(n) = T(n-1) + n^c$ $\quad c>0$, c is constant.

$T(n-1) = T(n-2) + (n-1)^c$

$T(n-2) = T(n-3) + (n-2)^c \rightarrow \underbrace{n^c + (n-1)^c + (n-2)^c \ldots}_{n \text{ terms}} 1^c = n.n^c = n^{c+1}$

$\vdots \qquad \vdots$

$T(1) = \underbrace{T(0)}_{\text{constant}} + 1^c$

$T(n) = T(n-2) + (n-1)^c + n^c$

$T(n) = T(n-3) + (n-2)^c + (n-1)^c + n^c$

$\ldots \quad \ldots$

$\Big\} \; T(n) = n^{c+1} = O(n^{c+1})$

i) $T(n) = T(n-1) + c^n$ $\quad c>0$ and c is constant

$T(n-1) = T(n-2) + c^{n-1}$

$T(n-2) = T(n-3) + c^{n-2}$ $\quad$ n terms $\; c^n + c^{n-1} + c^{n-2} + \ldots + c =$

$\vdots \qquad \vdots$

$T(1) = \underbrace{T(0)}_{\text{constant}} + c$

$T(n) = T(n-2) + c^{n-1} + c^n$

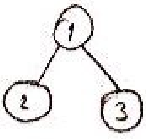
$T(n) = T(n-3) + c^{n-2} + c^{n-1} + c^n$

$c^n + c^{n-1} + c^{n-2} + \ldots c = \frac{c^{n+1} - 1}{c-1}$

$\lim_{n\to\infty} \frac{c^{n+1} - 1}{c-1} = c^n \qquad T(n) = O(c^n)$

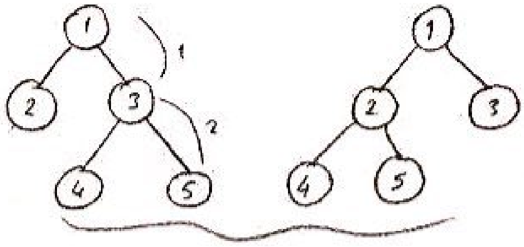
CamScanner ile tarandı

② a) B_3:



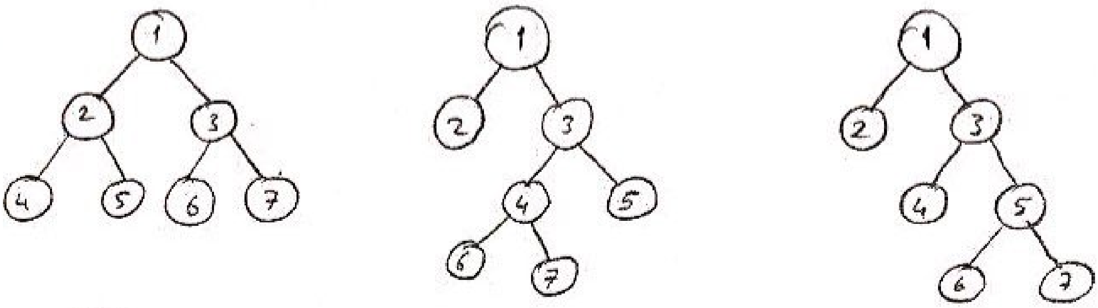
1

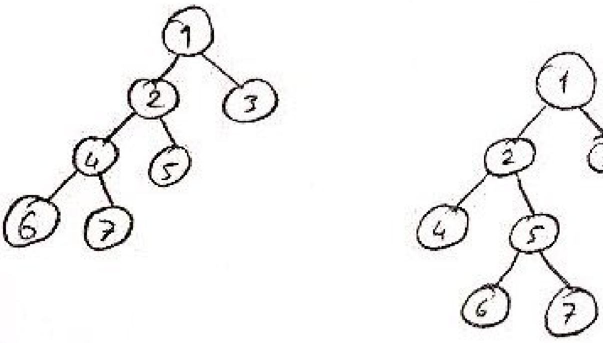
B_5 =



2

B_7 =





B_7 = 5 terms

We have a root node. In full binary tree, each nodes can have zero or two children, can not have one child. (when a node/vertex has one child, it is not a full binary tree.) When each nodes have two children, then number of vertices increases by two. And we have also a root node. The number of vertices must be odd numbers.

b) We can write the recurrence relation by decomposing the tree into subtrees. The tree has n nodes, number of nodes in left subtree $\underbrace{1, 3, 5, 7 \ldots n-2}_{B_i}$ and in right subtree $n-2, n-4 \ldots 1 \rightarrow B_{n-1-i}$

The recurrenc relation for $B_n = \sum_{i=1}^{n-2} B_i \cdot B_{n-1-i}$

c) $B_n \geq c.2^n$ for $\forall$ $n \geq n_0 > 0$

for $\forall$ odd values for k $1,3,5..k$ $B_k \geq c.2^k$

the recurrence relation for B_{k+2}

$$B_{k+2} = \sum_{i=1}^{k} B_i . B_{k+1-i} \qquad B_{k+2} \geq c2^i . c2^{k+1-i} . \frac{(k+1)}{2}$$

$$B_{k+2} \geq c.2^{k+1} . \frac{c(k+1)}{2}$$

$$\frac{c(k+1)}{2} \leq 2 \qquad B_{k+2} \geq c.2^{k+2}$$

3 a) Master Theorem

$a=7$ $b=3$ $d=2$ $\qquad a = b^d$ $\qquad T(n) = 7.T(\frac{n}{3}) + 2n$

$7 \overset{?}{=} 3^2$ $\quad 7<9$ $\qquad T(n) = \Theta(n^2)$

b) $T(n) = 2T(n-1) + n$

$T(n-1) = 2T(n-2) + n-1$

$T(n-2) = 2.T(n-3) + n-2$ $\quad \}$ n terms

$\vdots$

$T(1) = 2T(0) + 1$

$n + (n-1) + (n-2) .... + 1 = 2n$

$T(n) = 2.[2T(n-2) + n-1] + n \Rightarrow 2^{n-1}$

$$2^{n-1} + 2.n \overset{2^{n-1} > 2n}{\Longrightarrow} \Theta(2^{n-1})$$

c) $a=4$ $b=2$ $d=2$

$a = b^d$

$4 = 2^2$ ✓ $\qquad T(n) = O(n^2.\log n)$

(4)

function min-cut (Graph):
    Initialize contracted graph CG as copy of orginal graph.
    while (there are more than 2 vertices):
        pick a random edge (u,v) in the CG
        Merge u and v in to a single vertex
        update CG
        remove self loops.
    end while
    return cut represented by two vertices.

Average case :

c = number of edges in min-cut
m = total number of edges
n = total number of vertices

S_1 = event that one the edges
S_2 = 2^{nd} iteration

$P[S_1] = \frac{c}{m}$ → probablity ⟹ $P[S_1] <= \frac{c}{cn/2} = \frac{2}{n}$

$P[S_1'] = (1 - \frac{c}{m})$ → not to be probablity ⟹ $P[S_1'] >= (1 - \frac{2}{n})$

$P[S_1'] + P[S_2'] + P[S_3'] \ldots P[S_{n-2}'] = (1-\frac{2}{n}) * (1-\frac{2}{n-1}) * (1-\frac{2}{n-2} \ldots$

$(\frac{n-2}{n}) * (\frac{n-3}{n-1}) * \ldots \frac{2}{4} * \frac{2}{3} = \frac{2}{n*(n-1)} = \frac{1}{n^2} \in \Theta(\frac{1}{n^2})$

Best Case = If we have 3 vertex with 3 edges is (each vertex has 3 edges)

This time is $O(n)$

(5) function f(n):

```
res = 0
if n ≤ 1:
    res = 1
else:
    for i in range(n):
        res += f(i) * f(n-i-1)
print(res)
return res
```

} recurrence relation.

in loop n=3
i=0 f(0) * f(2)
i=1 f(1) * f(1)
i=2 f(2) * f(0)

in loop
n=4
i=0 f(0) * f(3)
i=1 f(1) * f(2)
i=2 f(2) * f(1)
i=3 f(3) * f(0)

$T(1)=1$

$T(2)=[T(0)*T(1)]+[T(1)*T(0)]=2*[T(1)*T(0)]=2$

$T(3)=[\underbrace{T(0)*T(2)}_{2}]+[\underbrace{T(1)*T(1)}_{1}]+[\underbrace{T(2)*T(0)}_{2}]=5$

$\vdots$

$T(n-2)=[T(n-3)*T(0)]+[T(n-4)*T(1)]+\dots+[T(1)*T(n-4)]+[T(0)*T(n-3)]=2*[T(n-3)+T(n-4)\dots+T(0)]$

$T(n-1)=[T(n-2)*T(0)]+[T(n-3)*T(1)]+\dots+[T(1)*T(n-3)]+[T(0)+T(n-2)]$

$=$

$T(n-1)-T(n-2)=2*T(n-2)+c$

$\boxed{T(n-1)=3\,T(n-2)}$

$T(1)=1$

$T(2)=3.T(1)=3^1$

$T(3)=3.T(2)=3.3.T(1)=3^2$

$\vdots$

$T(n)=3^{n-1}$ $T(n)=O(3^{n-1})$

Ahmet Yusa Telli
151044092