Graph PDF questions

1) A) only 1 connected component ✗
B) 2 connected P–R ✗
C) 2 connected Q–R ✗
D) 2 connected P–Q ✓

2) TD – DFT tree of undirected DES
TB – BFT tree of undirected DES

DFT (undirected) → no forward (like undirected) → no cross (like DFS undirected)

BFT (undirected) → no forward → no backward

Consider the graph

Here, both B & D are at depth 1 and they are connected by edge BD.

$|B-D| = 0$

So, $|i-j| = 1$ is not necessarily true

3) both true

4) MNOPQR
start M:
MNQR
so A wrong
B) NQMPOR
start N:
NQMO
so, B wrong
C) QMNROP
start Q:
QMNOP
so C wrong
D) POQNMR
start P:
POQNMR
✗ ✗ ✓ ✗ ✗ ✗

5) {1, 2, 3, 4, 5, 6, 7}

AB → 1, 3 → 20
x = 3:
CD → 3, 4 → 12
x = 6:
CD → 2, 3 → 6
x = 5:
CD → 1, 2 → 2

→ total = 40
→ But x & 2 can interchange places
→ So, 2 × 20 = 80

6) ✗

7) This can also be solved as,
Consider 3 components in DFS:

This will result in 3 DFT trees

The no. of vertices in each of these trees will result in total vertices = n

k_1 k_2 . . . . k_x → Tree edges

k_1+1 k_2+1 . . . . k_x+1 → vertices

(If I have k edges in a tree, I will have k+1 vertices) → tree, not graph

$k_1+1 + k_2+1 + \dots k_x+1 = n$

$(k_1+k_2+\dots k_x) + (1+1+\dots x \text{ times}) = n$

$k + x = n$

$x = n-k$

8) ✗

9) n vertices, k tree edges
DFT
→ If it directed or undirected

n = 5, k = 5 } → no. of components (connected) = 2

a) k > 5 ✗
b) k = 1 → 6 ✗
c) n-k = 1 → 7-5-1 = 1 ✗
d) n-k → 7-5 → 2 ✓
Take only if component
n = 5, k = 4 } n-k → 5-4 → 1 ✓

10) M N P R Q

11) DFS
n vertices, m edges
DFS(node):
visit[node] = 1
for edge e of node:
if (unvisited[other]) DFS(w)

This could also be a graph

O(n)

O(m+n)

Given in note

12) No of connected components
for (every vertex v)
if (unvisited) BFS(v) } v → n

BFS(s):
visit[s] = 1;
Q.enqueue(s)
while (Q.hasStuff):
for every adjacent w of Q.dequeue
if (unvisited):
visit
enqueue } m

O(m+n)

13) DFS(v)
{
visited[v] = 1
for (every adj of v)
if (visited[adj] != 1)
DFS(adj)
}

DF traversal:
→ recursion depth = 0

Start at the other node

14) 1) d[u] < d[v] for E(u,v)

E(u,v) = E(B,C)

(or)

d[u] < d[v] ✗

2) d[u] < f[v] ⇒ E(u,v) = E(B,A)

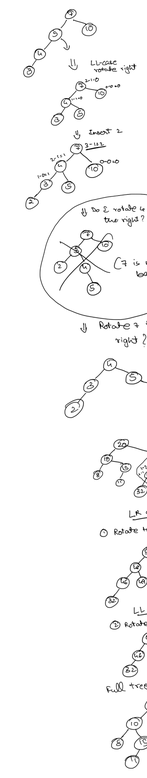
Vic C is traversed first

3) f[u] < f[v]

f[u] > f[v] ✓ | f[u] < f[v] ✗

4) f[u] > f[v] ✓

Inorder: 4 2 5 1 6 7 3
Postorder: 4 5 2 7 6 3 1



Tree PDF Question

1) 2-ary tree
X = Internal nodes
L = Leaf
L = ?
n = L + X
X * 2 + 1
L + X = X * 2 + 1
L = X + 1
L + X = X * k + 1
L = k(X-1) + 1

2) n, inter nal nodes
n * k + 1 = n + L
L = k(n-1) + 1

3) $n_2 = 10$, $n_1 = 5$
$n = n_2 + n_1 + n_0$
$n = 2 * n_2 + 1$
$n_0 = n_2 + 1$
⇒ $n_0 = 11$

4) $n_1 = 5$, $n_2 = 10$, $n_0 = ?$
$n = n_1 + n_2 + n_0$
$n = 2 * n_2 + 1$
$n_0 = n_2 + 1$
$n_0 = 10 + 1$
$n_0 = 11$

5) Preorder: ABC
Postorder: BCA
Based on pre-order & postorder, we get only 1 tree. So, only 1 tree is possible, only 1 tree is possible

6) Inorder with preorder and inorder with postorder can uniquely identify a tree. So, 2 & 3

7) Inorder: [illegible]
Preorder: abdcefg

Postorder: dbfegca

8) Re-solving question 5 (got wrong above)
Any 2 traversals:
In & Pre → 1
In & Post → 1
Pre & Post → not (no structure) unique
Pre & Post (binary tree) → 1

9) Inorder: ABC
Preorder: ABCD
Pre: ABCD
Last Post = root
Last Pre = rightmost node at last level
Last In =
In: BAC
Pre: ABC
Post: BCA
In = Post ✗
Pre = Post ✗
This is considering that complete binary tree is not 2-ary tree with leaf nodes at n & n-1 levels
In = Post ✗
Pre = Post ✗
In = Pre ✗
} not always true
In: BADCE
Pre: ABCDE
In: BADC
Pre: ABCD
?? What is the answer??
Did not solve completely.

10) 1 M B C A F H P Y K
2 K A M C B Y P F H
3 M A B C K Y A P H
Preorder – root first
Post – root rear
1 is post
2 is pre
3 is in

11) 3 4 * 5 - 2 ^ 6 7 * 1 + -
3 4 * → 3*4 = 5
2 5 - → (3*4 - 5)
6 7 * → 6*7
6 7 * 1 + → 6*7+1
⇒ (3*4 - 5)^2 - (6*7+1)
- + 1 * 7 6 ^ 2 - 5 * 4 3
SAW THE SOLUTION FOR THIS

10) Taking out no. time to think on this
n = 3, no. of possible BSTs for distinct keys
⇒ $\frac{{}^{2n}C_n}{n+1}$ ⇒ $\frac{{}^{6}C_3}{4}$
⇒ $\frac{6!}{3! \times 3! \times 4}$ ⇒ 5
2 leaf nodes
Total 4 BSTs possible

11) 1, 2, 3, . . . n
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
First member needs to be 8-1 = 7
n = 10 ⇒ 10-3
⇒ 7
So, first member to be inserted should be n-P

12) ⇒ 0
⇒ 1
⇒ 2
⇒ 3
Lowest level is level 0 or level 3?
Based on options, it is level 3.
So, 67
Why did I think this??

13) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
⇒ 55 ✗ 4 > 4.7
Did this carefully but missed it while writing it down

14) Diff in heights
⇒ |left − right| = 2
For minimum no. of nodes, let's take at max 2 nodes at each level (one for the left side & the other for the right side)
For minimum no. of nodes, one side needs to have at least height no. of nodes and the other side atleast height − 2
Eg: height = 4
no. of nodes ⇒ height
no. of nodes = 2 height − 2
For height = 8
h = 1 + (h−2)
(left) (root)
⇒ 8 + 1 + 6
⇒ 15
Did not read question thoroughly

15)

Tree questions PDF:

Pg 23 Q) n(0) = 1
n(1) = 2
n(2) = 3 + 1 = 4
n(3) = 4 + 3 + 1 = 8

Pg 25 Q) In max heap,
root is max value
And division:
2 * i + 1
2 * i + 2

a) root max heap
b) root max heap
c)
d)

Pg 26 Q) Heap is a CBT
left to right
structure possible: 1

1 2 3 4 — Fixed
4!
⇒ 24

x & y cannot be 6 & 7

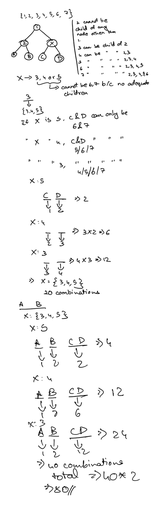
x & y can be any of the numbers other than 1, 6 & 7
3 4 ⇒ 12
1 2 3 4 ⇒ 4! ⇒ 24

x & y cannot be 6 & 7 bec their children would need to be greater than root

So, if x is 5, its children can only be 6 & 7
If x was 2, its children could be any 2 out of 3, 4, 5, 6 & 7
x = 5: 2! (1 2) {6, 7} & {6, 7}
x = 4: 6 (2 3)
x = 3: 12 (3 4)
x = 2: 20 (4 5)
x can have a total of 40 combinations
Same for y
⇒ 40 + 40 ⇒ 80

Pg 27 Q)
a)
b)
c)
d)

Pg 28 Q)
A) ✗
B)
C)
D)

Pg 29 Q)
40, 35, 20, 10, 30, 16, 17, 8, 4, 15

Pg 30 Q) In min heap, node is less than all its children and their subtree.
$(2^{h+1}) - 1 = 1023$
$2^{h+1} = 1024$
h + 1 = 10
h = 9
Max depth: 9
P < 9
P ∈ {1 . . . 8}
If P != 1 → has another parent P_2