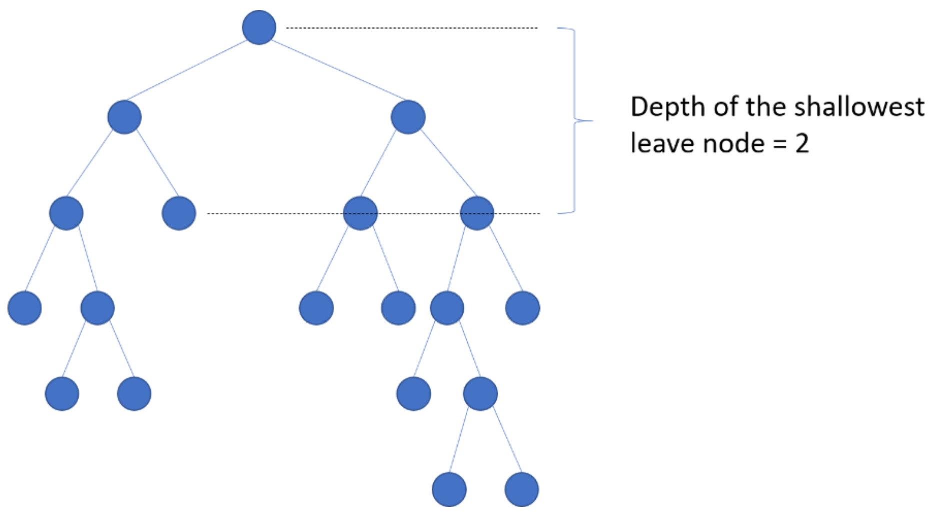
1. Given a binary tree as an input, design a recursive tree algorithm to find the <u>depth of the shallowest leaf node.</u>



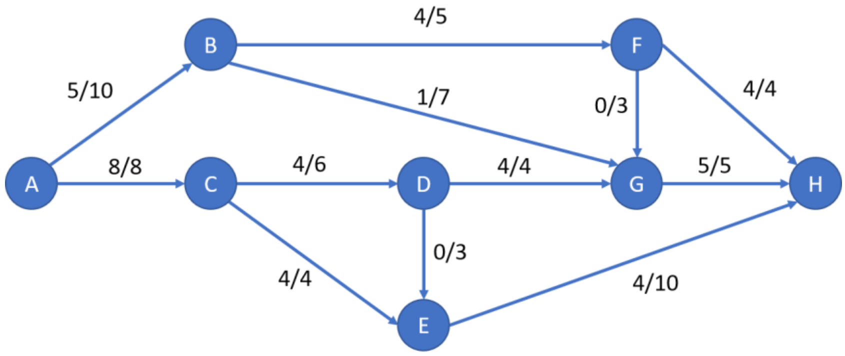
1.1 Task assigned to the left and right subtree

1.2 The output from the left and right subtrees

1.3 The extra work to process the output from the left and right subtree

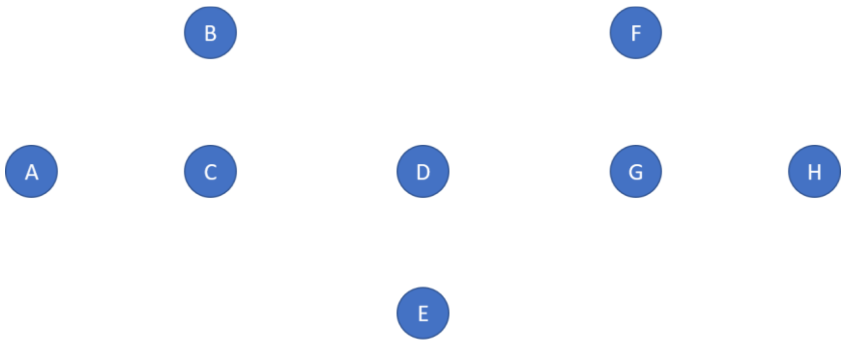
1.4 Input and output for the simplest case

2. Consider the following flow network



2.1 What is the total flow in the current stage?

2.2 Draw the residual graph of the current stage



2.3 Find an additional path from the residual graph in the previous step in terms of node sequences and the maximum capacity of that path.

2.4 By using “the max-flow min-cut theorem”, find a cut to confirm that the maximum flow is found.

3. Some kind of “jumping across the river” analysis (not done yet)

4. Find a counterexample to show that greedy algorithm for the event scheduling problem does not guarantee to return the optimal solution if we use the priority function as follows:

1.1 Shortest event first

1.2 Highest number of conflicts first

5. The greedy coin payment algorithm pays a positive integer amount of money with the largest value coin that fit the remaining value first. We expect that the algorithm make a payment with a fewest number of coins.

- The coin values are [1, 2, 5, 10]
- Assume there are unlimited number of coins

Given the source code implementing the coin payment algorithm as follows:

However, the code is not correct, so that the number of coins is not optimal.

Rewrite some lines of codes to make it work.

Line of code	Old code	New code

6. Find the longest common subsequence of S_1 and S_2 by using the dynamic programming algorithm.

$S_1 =$ [T, R, A, I, N, E, R]

$S_2 =$ [T, R, A, D, E, R]

a. Fill the table of longest common subsequence size

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								

b. Trace for the subsequence by drawing circles around the objects that has been matched, started from the bottom right corner of the table.

7. Find the optimal cost for the following matrix chain multiplication

$$A_{2\times3}B_{3\times5}C_{5\times3}D_{3\times6}E_{6\times2}$$

a. Fill in the optimal cost table and the reference table

Optimal cost table

	A	B	C	D	E
A					
B					
C					
D					
E					

Reference table

	A	B	C	D	E
A					
B					
C					
D					
E					

7.3 Write the optimal solution in the form of nested parenthesis of multiplications.

8. Assume that there are unlimited number of processors. Parallel quick sort can be done by assigning every individual processor to do a partitioning task in the same level.

8.1 Fill the table

Level	Sub-instance size	Work in a stack frame	Number of frames	Total work
0				
1				
2				
...	...	...	...	...

8.2 Calculate the time complexity of the parallel quick sort