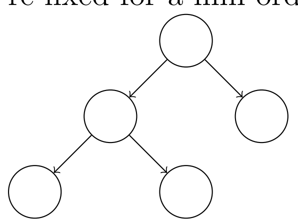
COSC 202, Exam 3 Practice

The solutions are not intended to be detailed. On the exam you need to provide answers in sufficient detail to get full credit. If you are uncertain what counts as sufficient detail for any particular question, please ask your instructor.

1. Suppose you start with an empty min-ordered heap and perform the following sequence of operations in the following order: insert 20, insert 10, insert 5, insert 25, insert 30, insert 35, remove min, insert another 30, and finally remove min.
   Give the location of each item in the resulting heap by filling in the nodes:
   Hint: After each insert or remove and before the next operation, the heap should be re-fixed for a min-ordered heap.



2. Given an array C[ ] of size m and a target value X, the goal is to find the minimum number of coins required to reach the target X. For example, if C = [25, 10, 5] and X = 30, then the minimum number of coins is 2 (one coin of denomination 25, and one of denomination 5). Similarly, if C = [9,6,5,1] and X = 13, minimum number of coins is 3 (two coins of 6, one coin of 1). If it is not possible to get to a value X given C, then the code should return ∞. For example if C = [10,5,2] and X = 3, then it is impossible to get to that value given the coins we have, so it should return ∞.

   (a) Write the recurrence relation for MinCoin[X].

(b) Formulate a recursive algorithm that implements the above recurrence relation.

```
int minCoin(int C[], int m, int X)
   Your code goes here
    return min
```

(c) Formulate a Dynamic Programming implementation of the recurrence relation by completing the following code:

```
int minCoin(int C[], int m, int X)
    int coinChange[X+1]
    coinChange[0] = 0
    Your code goes here
    return coinChange[X]
```

(d) For the coin denomination of $C = [2, 5, 7]$ and $X = 10$, trace your code and find the minimum number of coins. Please show intermediate steps by filling the coinChange table below:

value	0	1	2	3	4	5	6	7	8	9	10
min coins											

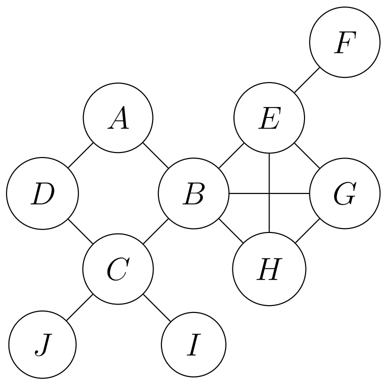
(e) What is the time and space complexity of the DP implementation in terms of X and m?

3. Recall the **pretty print** problem: Given an input vector representing lengths of n number of words: $W = [w_0, w_1, \ldots w_{n-1}]$, and an input integer representing maximum length L, we are to minimize sum of number of spaces at the end of each line. i.e., the penalty function is linear, $f(x) = x$. A greedy strategy fits as many words possible in the first line and repeats the process for other lines.

   (a) What is the runtime of the above greedy algorithm with respect to n and L?

   (b) When $f(x) = x$, does the above greedy algorithm always reach optimum results? If yes, justify correctness. If no, describe a counter example.

4. Consider the following connected, undirected, and unweighted graph. Suppose we start a Breadth-First Search (BFS) from vertex A as the source. Draw the resulting tree.



Hint: You can use a queue to keep track of vertices. Break ties alphabetically.

queue										

5. You are given a large database of webpages on the internet and your goal is to assign a score for how popular each page is. You can define popularity of some page x in terms of the number of webpages that reference x.

   (a) How could you encode the information in the database as a graph? What are the nodes and what are the edges?

   (b) Describe how you could compute the popularity of each webpage. Analyze the time complexity.