Design and Analysis of Algorithm (CS 412)
Instructor: Dr. Ayesha Enayet
Date: _______________
CS 6th
SIS ID:________________ Name:_________________

Note: Attempt all the questions. Use blue or black pen only.

1. **Write down an algorithm for finding a minimum spanning tree of an unconnected graph.** [1]

Call MST_KRUSKAL(G,w)

2. **Which algorithm, dynamic programming or greedy, is better suited for the fractional knapsack problem? Justify your choice with a time complexity analysis. [Note: you must give a valid justification for your answer to score a point. No partial marking]** [1]

Greedy approach is better suited for fractional knapsack problem. The worst case time complexity of greedy approach is O(nlgn) while constructing the dynamic programming table would require considering all possible combinations of items and their fractions, which would lead to an exponential number of subproblems.

3. **Write a dynamic programming algorithm to find an array's longest increasing subsequence (LIS). The elements of the subsequence should be contagious. For example, <3,6,7> is the LIS of an array {2,4,3,6,7,1}.** [1]

```
I.   Initialize an array LIS of size n.
II.  Initialize all elements of LIS to 1 (Base case).
III. Iterate over input array A:
        a. If A[i]>A[j]: [i is current element and j is the previous element of the array A]
              i. LIS[i]=LIS[j]+1
IV.  Return max(LIS)
```

4. **Determine the Huffman code for each alphabet of a file based on their frequencies as provided in the following table.** [0.5]

Char	a	b	c	D	e	F
Frequency (Thousand)	20	10	5	50	14	1
Variable Codeword	00	0111	01101	1	010	01100

a. **Calculate the number of bits required to represent the file using fixed length codeword.** [0.25]

100000*3=300,000

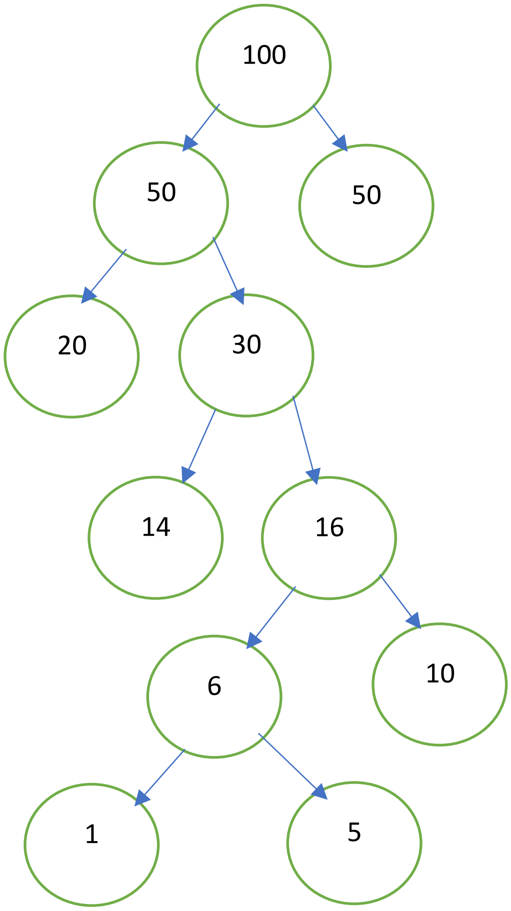
b. **Calculate the number of bits required to represent the file using a variable length codeword (Huffman code)**[0.25]

20*2+10*4+5*5+50*1+14*3+1*5=202000

5. **Compare the time complexities of Dijkstra's and Bellman-Ford's algorithms and determine which algorithm is more efficient in terms of time complexity.** [1]

The time Complexity of Dijkstra's is O((V + E) log V) with a binary heap or Fibonacci heap. The time Complexity of Bellman Ford is comparatively high O(VE).

Char	a	b	c	D	e	F
Frequency (Thousand)	20	10	5	50	14	1
Variable Codeword	00	0111	01101	1	010	01100


100
50
50
20
30
14
16
6
10
1
5