Data Structure Final CPT

1. Given a directed graph G where each node represents an activity labeled from A-Z, output the topological sort order. When there are multiple nodes with in-degree = 0, you must choose the node with the label that has the smallest alphabetical order first.
2. Given a directed graph G, output the in-degree and out-degree for each node.
3. Given a directed graph G, output the transitive closure in the form of an adjacency matrix
4. Implement quicksort and print out the intermediate sorted result for each pass.
5. Given a weighted directed graph, implement Kruskal's algorithm and output the minimum cost spanning tree.
6. A depth-first-search question (the magic pocket)

   There is a magic pocket, and the capacity is k. You can use this pocket to get any number of items and the total cost of these items must be equal to k. The sizes of all items are a_1, a_2, …, a_n, respectively. Find the number of all possible combinations.

   The input contains two lines. The first line contains two integers k (capacity) and n (total number of items); the second line consists of n integers that represent the item sizes a_1, a_2, …, a_n.

   The output is the number of all possible combinations.

   For example:

   <Input>

   40 3

   20 20 20

   <Output>

   3
7. A breath-first-search question (a strange lift)

   The strange lift can stop at every floor as you want, and there is a number Ki (0 <= Ki <= N) on every floor. The lift has just two buttons: up and down. When you at floor i and if you press the button "UP", you will go up Ki floors. That is, you will go to the the (i+Ki)-th floor. If you press the button "DOWN", you will go down Ki floors. That is, you will go to the (i-Ki)-th floor. The lift can't go up high than N, and can't go down lower than 1. For example, there is a buliding with 5 floors, and k1 = 3, k2 = 3, k3 = 1, k4 = 2, k5 = 5. Beginning from the 1st floor, you can press the button "UP", and you'll go up to the 4th floor, but if you press the button "DOWN", the lift can't do it, because it can't go down to the -2th floor. Given two parameters A and B, where A represents the floor you are located and B is the floor you want to go, print the least total number of "UP" and "DOWN" to reach the target floor. If you can not reach floor B, print "-1". The input contains two lines. The first line contains three integers N, A, B, whrer N is total of floors in the building (N >= 1, 1<= A, B <= N); the second line consists of N integers k1, k2,…, kn.

For example:

<Input>

5 1 5

3 3 1 2 5

<Output>

3

8. Find the shortest path using the Dijkstra algorithm. Show the cost of the shortest path between two specified vertices in the weighted directed graph.
9. Implement Prim's algorithm. Print out each edge of the minimum cost spanning tree for each step during the process.
10. Implement bubble sort and print out the intermediate sorted result for each pass.