Design and Analysis of Algorithms

Exercises 4/8

1. Estimate the time complexity of Prim's algorithm if array is used as a data structure for the set. Demonstrate how the data structure is used when solving the problem instance below.

2. *Binary Knapsack* problem is defined using binary selections: we must make decision for every item either take or leave it. Consider the following two greedy Algorithms. Algorithm A always selects the costliest (c_i) item in the knapsack that still fits in. Algorithm B always selects the one with most valuable (cost per weight = c_i/w_i). Give two counter examples when **(a)** algorithm B finds the optimal solution, but algorithm A fails, **(b)** they both fail.

3. Design a greedy algorithm for open loop *Travelling Salesman Problem* (TSP). Apply the algorithm for the graph shown in Task 1 starting from three different target nodes. Draw the tours and calculate their total length. Based on your experiments, design a heuristic choice for selecting the start node for the algorithm. Does the algorithm provide the optimal result?

4. Show that the cost of its minimum spanning tree is always smaller than the cost of the open loop travelling salesperson problem for the same graph. Can you design a bounding criterion for branch-and-bound TSP algorithm using this property?

5. Suppose that a good bouncing criterion can cut of 50% from the search tree at the root node. How much faster it is than full search? How much it improves the time complexity? What if 50% cut off can be done at every level in *every node,* how much faster it makes the algorithm and time complexity?