Homework Assignment 10

Due Date: May 20, 2022, 23:59

Exercise 1. Suppose the edges of a graph are presented to you only once (for example over a network connection) and you do not have enough memory to store all of them. The edges do not necessarily arrive in sorted order.

Outline an algorithm that nevertheless computes a minimum spanning tree using space in $O(|V|)$.

total points: 8

Exercise 2.

(i) In some instances of the greedy TSP algorithm it is possible to find a shorter optimal tour if the salesman is allowed to visit a city more than once. Construct an example for this.

(ii) A distance matrix is *Euclidean* iff it satisfies the triangle inequality: $d(i,j) \leq d(i,k) + d(k,j)$ for all i, j, k. Show that in this case it is never advantageous to pass through the same city several times.

(iii) Specify a greedy algorithm for the TSP with a Euclidean distance matrix, which produces tours with at most twice the length of an optimal tour.

Hint: Start by constructing a mimimal spanning tree, then use (ii).

total points: 12

Exercise 3.

(i) The Ford-Fulkerson algorithm when executed in a depth-first fashion has some redundancy. First, all outgoing edges are pushed onto the stack and then the last is popped off to be followed by the algorithm. Modify the Ford-Fulkerson algorithm such that the first edge coming out of a certain vertex is immediately followed, and the second is followed only if the first does not lead to the sink. Consider using recursion.

(ii) The Ford-Fulkerson algorithm assumes that it terminates. Do you think such an assumption is safe?

total points: 10

Exercise 4.

(i) Implement Kruskal's algorithm for the construction of a minimum spanning tree.

(ii) Implement a *partition data structure* (aka *union-find data structure*) to support your implementation in (i). For this use a list L, in which entries correspond to vertices. If $L[i] = j$, then vertex i is in the connected component of j. In this way each connected component is represented as a tree, so can easily check if the endpoints of the current edge under consideration are already in the same connected component. Show how to realise the merging of connected components.

total points: 20