DESIGN AND ANALYSIS OF ALGORITHMS — HT 2019

Problem Sheet 4

Questions marked with $*$ are not intended to be discussed in tutorials, answers to these questions will be posted on the course webpage.

DFS and connected components, cont'd

Question 1

Design a linear-time algorithm that, given an undirected graph G and a particular edge e in it, determines whether G has a cycle containing e.

Question 2

Design a linear-time algorithm that, given a directed graph G, detects if there is an odd-length cycle. (*Hint*. First solve the problem under the assumption that the graph is strongly connected.)

Shortest paths

Question 3

$*$ It is easy to see that for any graph BFS and DFS will take almost the same amount of time. However the space requirements may be quite different.

(a) Give an example of an n-vertex graph for which the depth of recursion of DFS starting from vertex v is $n-1$ whereas the queue of BFS will have at most one vertex at any given time if BFS is started from v.

(b) Give an example of an n-vertex graph for which the queue of BFS will have $n-1$ vertices at one time whereas the depth of recursion of DFS is at most one. Both searches being started from the same vertex.

Question 4

Often there are several shortest paths between two vertices of a graph. Give a linear-time algorithm for the following task:

***Input*:** An undirected graph $G = (V, E)$ with unit edge lengths; vertices $u, v \in V$.

***Output*:** The number of distinct shortest paths from u to v.

Question 5

Give an algorithm that takes as input a directed graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O((|V|^2 + |V||E|) \log |V|)$.

Greedy algorithms

Question 6

$*$ Assume an unlimited supply of coins of each denomination. For which of the following *change making problems* does the greedy algorithm always find an optimal solution? Justify your answer.

(a) Each coin in the series is worth at least twice the next lower denomination, and the series includes a 1-unit coin.

(b) The available denominations are $p^k, p^{k-1}, \cdots, p, 1$ where $p > 1$ and $k \geq 0$.

(*Hint.* You may find the inequality $p^k > (p-1)(p^{k-1} + \cdots + p + 1)$ useful.)

Question 7

(a) Show that if T is a spanning tree for the undirected graph G, then the addition of an edge e, $e \notin E(T)$ and $e \in E(G)$, to T creates a unique cycle.

(b) Show that if any one of the edges on this unique cycle is deleted from $E(T) \cup \{e\}$ then the remaining edges form a spanning tree of G.

Question 8

(a) Prove that if the edge weights of an undirected weighted graph are all distinct, then it has a unique minimal spanning tree.

(b) Show how to find the *maximal* spanning tree of a graph, that is, the spanning tree of the largest total weight.

Question 9

Each member of a group of deep-sea divers, $d_1, \ldots, d_n$, is on the sea bed. The divers have to return to the surface but only one of them can be lifted at a time. It takes time t_i to lift diver d_i to the surface.

(a) In what order should the divers be lifted so that the sum of the times spent underwater by the divers is minimized?

(b) Can you prove that your answer is correct?

(*Hint*: consider an order different from the one believed to minimize the sum of times, and show that an order with a smaller sum of times can be obtained by swapping the order of two divers.)

(c) Now suppose that each diver d_i has enough air to stay under water only for time a_i. An order for lifting the divers is *safe* if each d_i reaches the surface no later than time a_i.

i. Assuming that $a_1 \leq \ldots \leq a_n$, show that if there is a safe order then the order $d_1, \ldots, d_n$ is safe.

ii. Hence design an efficient algorithm that, given $t_1, \ldots, t_n$ and $a_1, \ldots, a_n$, determines whether there is a safe order.