FTP_Alg_Week 6: Exercises

jungkyu.canci@hslu.ch

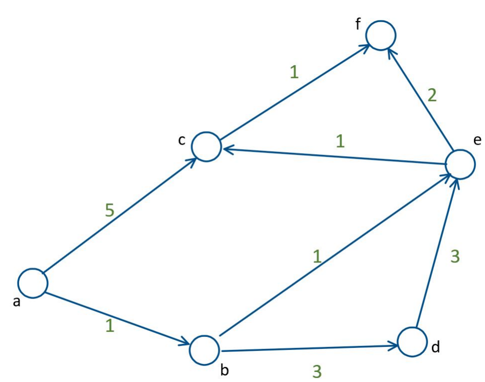
21 October 2024

Exercise 1 *The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \mid (u, v) \in E\}$. Describe efficient algorithms for computing G^T from G, for both the adjacencylist and adjacency-matrix representations of G. Analyze the running times of your algorithms.*

Exercise 2 *(Optional) Give an example of a directed graph $G = (V, E)$, a source vertex $s \in V$, and a set of tree edges $E_\pi \subset E$ such that for each vertex $v \in V$, the unique simple path in the graph (V, E_π) from s to v is a shortest path in G, yet the set of edges E_π cannot be produced by running BFS on G, no matter how the vertices are ordered in each adjacency list.*

Exercise 3 *Give a simple example of a directed graph with some negative-weight edges for which Dijkstra's algorithm produces incorrect answers.*

Exercise 4 *We apply DIJKSTRA (Lecture 11 and Lecture 12) to the graph (G, V) represented in the following picture:*



Dijkstra is an iterative procedure, which update at each step the value $v.d$, that is the distance of the node v to the root a. After INITIAL-SINGLE-SOURCE(G,a) we have $a.d = 0$ and $v.d = \infty$ for each vertex $v \neq a$. We consider the situation after two iterations (line 4 to line 8) of Dijkstra with *starting node a(Look out! We consider only two iterations and not the whole Dijkstra's procedure).*

What is c.d?

What is f.d?

What is d.d?