Discrete Optimization 2010

Hand-in Assignments (due Oct. 04, 2010)

Marc Uetz (`m.uetz@utwente.nl`), Applied Mathematics, University of Twente

Read Chapter 8 of [1] about Matroids and the greedy algorithm. Read Sections 25.1-25.3 and 26.2 of [2] for shortest path algorithms. Read Section 3.4 of [3] for graph search algorithms like BFS and DFS.

1. (5 points) Given an undirected graph $G = (V, E)$, a *forest* F is a subset of the edges that does not induce a cycle in G. (Notice that a forest need not be connected; it can consist of up to $|V|$ components.) Prove that the system $M = (E, \mathcal{I})$ with $\mathcal{I} = \{F \subseteq E \mid F \text{ is a forest in } G\}$ is a matroid.

2. (6 points) Given is a directed graph $G = (V, A)$ with arc lengths c_{vw} for $(v, w) \in A$. Let $s \in V$, and assume there is a directed path from s to all $v \in V \setminus \{s\}$.

   (a) Let $d(v)$ denote shortest path lengths from s to all other nodes $v \in V$. Show that the *Bellman equations* hold, that is,

   $$d(w) = \min\{d(v) + c_{vw} \mid (v, w) \in A\} \quad \forall\, w \neq s$$

   (b) Show that any set of labels $d(v)$ that satisfy the Bellman equations are shortest path lengths, as long as G has no cycle of length ≤ 0.

   (c) Give an example of a digraph and node labels $d(v)$ that satisfy the Bellman equations, but do *not* represent shortest path lengths.

3. (5 points) Given an *acyclic* and connected digraph[1] $G = (V, A)$, $s \in V$, and arbitrary arc lengths c_a, $a \in A$. Assume $n = |V|$ and $m = |A|$. Let us assume that a linearization (topological order) π of G is given[2]. Describe (in simple, high-level pseudo-code) an algorithm for finding shortest (s, v)-paths for all $v \in V$ in time $O(m)$. Prove correctness and the linear bound on the computation time.

4. (not for delivery) Think about an instance of a digraph $G = (V, A)$ with negative arc lengths, but without negative cycle, such that Dijkstra's algorithm fails (i.e., does not compute the correct shortest path lengths). Why can't we just preprocess the graph and add a large enough constant to all arc lengths to guarantee they are all nonnegative?

References

[1] Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, Wiley-Interscience, 1998.

[2] Cormen, T. H., C. E. Leiserson, and R. L. Rivest (1990), *Introduction to Algorithms*, MIT Press.

[3] Ahuja, R. K., T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice Hall, 1993.

[1] Here, connected simply means that the underlying undirected graph is connected.

[2] A linearization of a connected digraph can be computed in $O(n + m)$=$O(m)$ time; see literature.