BU CLA CS 530: Analysis of Algorithms

Spring 1996

Problem Set 4

Due Thursday 96.02.15, at 11:00 am, in class. Late submissions not accepted.

---

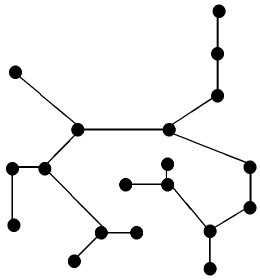
1. Exercise 23.4-3, page 488, in [CLR].

2. Let $G = (V, E)$ be a connected undirected graph, with weight function $w : E \to \mathbf{R}$. Let F be a subgraph of G which is a forest, i.e. F does not contain cycles and consists of zero or more trees. Design an efficient algorithm to construct a spanning tree T of G which contains all the edges of F *and* which minimizes the weight among all spanning trees containing F. *Hint*: Consider one of the two algorithms proposed in Chapter 24 in [CLR]. For a complete solution you should also prove your algorithm correct.

3. Exercise 25.2-4, page 531, in [CLR].

4. Exercise 25.3-3, page 535, in [CLR]. You do not know m in advance, so you cannot just make the algorithm loop m times (instead of $|V| - 1$) and then stop.

5. Let $G = (V, E)$ be a connected undirected acyclic graph (so G is just a tree) where every vertex has degree $\leqslant 3$. G may therefore look like



   Let $n = |V|$. Show that G has an edge whose removal disconnects G into two disjoint subgraphs with no more than $(2n+1)/3$ vertices each. Give an algorithm that finds such an edge in linear time, i.e. the algorithm visits the vertices and edges of G at most $\mathcal{O}(n)$ times. *Hint*: Consider a depth-first search through G to determine, for each $e \in E$, the sizes of the two connected components obtained by deleting e. Orient the edge e in the direction of the smaller component. G thus becomes directed. Then .... In the resulting directed tree, every vertex has in-degree $\leqslant 1$ (why?) and exactly one has in-degree $= 0$ (why?).

Created: 96.02.08
Modified: 96.02.12