CS 473: Undergraduate Algorithms, Spring 2009
Homework 6½

Practice only—do not submit solutions

1. In class last Tuesday, we discussed Ford's generic shortest-path algorithm—relax arbitrary tense edges until no edge is tense. This problem asks you to fill in part of the proof that this algorithm is correct.

   (a) Prove that after *every* call to Relax, for every vertex v, either $\mathit{dist}(v) = \infty$ or $\mathit{dist}(v)$ is the total weight of some path from s to v.

   (b) Prove that for every vertex v, when the generic algorithm halts, either $\mathit{pred}(v) = \textsc{Null}$ and $\mathit{dist}(v) = \infty$, or $\mathit{dist}(v)$ is the total weight of the predecessor chain ending at v:

   $$s \to \cdots \to \mathit{pred}(\mathit{pred}(v)) \to \mathit{pred}(v) \to v.$$

2. Describe a modification of Shimbel's shortest-path algorithm that actually computes a negative-weight cycle if any such cycle is reachable from s, or a shortest-path tree rooted at s if there is no such cycle. Your modified algorithm should still run in $O(VE)$ time.

3. After graduating you accept a job with Aerophobes-Я-Us, the leading traveling agency for people who hate to fly. Your job is to build a system to help customers plan airplane trips from one city to another. All of your customers are afraid of flying (and by extension, airports), so any trip you plan needs to be as short as possible. You know all the departure and arrival times of all the flights on the planet.

   Suppose one of your customers wants to fly from city X to city Y. Describe an algorithm to find a sequence of flights that minimizes the *total time in transit*—the length of time from the initial departure to the final arrival, including time at intermediate airports waiting for connecting flights. *[Hint: Modify the input data and apply Dijkstra's algorithm.]*