Practice exercises for midterm

do not return

Exercise 1

Suppose that you have two sets of vertices S and T which are both subsets of vertex set V of a directed graph. The graph may have negative edge weights. Suppose that you want to find the shortest path with the property that it starts at a vertex in S and ends at a vertex in T.

Of course, you can solve this problem by running the Bellman-Ford algorithm $|S|$ times. This algorithm would run in $\mathcal{O}(mn|S|)$ time. But you can do better. Describe an algorithm that solves this problem in $\mathcal{O}(mn)$ time, i.e., the time needed by a single run of the Bellman-Ford algorithm.

Exercise 2

Suppose you have a hash table with $m = 2^{20}$ entries that should store elements with 32 bit keys. Show that the family of hash functions

$$\mathcal{H}_t = \left\{ h_t : t \in \{0..m-1\}^{\left\{0..2^{12}-1\right\}} \right\}$$

where

$$h_t(x) = (x \ mod \ 2^{20}) \oplus t[x \ div \ 2^{20}]$$

is universal for this case.

Exercise 3

We are given a graph describing the streets of a (hypothetical) American city. All streets go either east–west or north–south. There are no oneway streets and no turning restrictions. A limousine service wants to have a navigation system for stretch limousines that finds the fastest route between two given street intersections. The required time is the length of the route *plus* $C \cdot k$, where k is the number of turns on the route. The constant C gives the time it takes to make a turn.

Give an algorithm that solves the problem in $O(n \log n)$ time. Here $n = |V|$ is the number of street intersections in the city.

Exercise 4

Outline a linear time algorithm for computing single source shortest paths in graphs that have strongly connected components with at most 42 nodes. Your algorithm should work for general edge weights.