Homework Assignment 3

Due: Friday, December 11

Q1: **Connectivity**

An undirected graph is *biconnected* if any pair of distinct vertices lie on a simple cycle (no repeated vertices).

a) Give a linear time algorithm to compute maximal biconnected components by using DFS.

An undirected graph G is k-*connected* if between any pair of distinct vertices there are k edge disjoint paths, and it is *precisely* k-connected if it is k-connected and not $(k+1)$–connected.

b) Give an efficient algorithm to determine for what k is a given input graph precisely k-connected.

An undirected graph G is k-*edge separable* if there exists a set of k edges whose removal disconnects the graph into two parts, and it is *precisely* k-edge separable if it is k-edge separable and not $(k-1)$-edge separable.

c) Prove or disprove: G is precisely k-edge separable if and only if it is precisely k-connected.

Q2: **Edit distance**

Given a string $s \in \Sigma^*$, where Σ some finite alphabet, it is possible to perform the following string edit operations:

- $\mathsf{insert}(s, i, \alpha)$ returns the string $s[1..i-1]\alpha s[i..|s|]$, where $1 \le i \le |s|+1$ and $\alpha \in \Sigma$.
- $\mathsf{delete}(s, i)$ returns the string $s[1..i-1]s[i+1..|s|]$,
- $\mathsf{substitute}(s, i, \alpha)$ returns the string $s[1..i-1]\alpha s[i+1..|s|]$.

The *edit distance* $\delta(s,t)$ between two strings s and t is the minimal number of edit operations required to transform s to t. The edit distance $\delta(s,t)$ can be computed by the following recursive formula:

$$\delta(s,t) = \begin{cases} |s| & \text{if } \ |t| = 0 \\ |t| & \text{if } \ |s| = 0 \\ \min\{1+\delta(s, t[2..|t|]), 1+\delta(s[2..|s|], t), 1+\delta(s[2..|s|], t[2..|t|])\} & \text{if } \ s[1] \neq t[1] \\ \min\{1+\delta(s, t[2..|t|]), 1+\delta(s[2..|s|], t), \delta(s[2..|s|], t[2..|t|])\} & \text{if } \ s[1] = t[1] \end{cases}$$

a) Show how to compute $\delta(|s|, |t|)$ in time $O(|s| \cdot |t|)$ and additional space $O(\min\{|s|, |t|\})$.

b) Show how to compute a sequence of edit operations of length $\delta(|s|, |t|)$ which transforms s to t in time $O(|s| \cdot |t|)$ and additional space $O(\max\{|s|, |t|\})$.

Q3: **Ghostbusters**

A group of n Ghostbusters is battling n ghosts. Each Ghostbuster is armed with a proton pack, which shoots a stream at a ghost, eradicating it. A stream goes in a straight line and terminates when it hits the ghost. The Ghostbusters decide upon the following strategy. They will pair off with the ghosts, forming n Ghostbuster-ghost pairs, and then simultaneously each Ghostbuster will shoot a stream at his or her chosen ghost. As we all know, it is *very* dangerous to let streams cross, and so the Ghostbusters must choose pairings for which no streams will cross.

Assume that the position of each Ghostbuster and each ghost is a fixed point in the plane and that no three positions are collinear.

a) Argue that there exists a line passing through one Ghostbuster and one ghost such the number of Ghostbusters on one side of the line equals the number of ghosts on the same side. Describe how to find such a line in $O(n \log n)$ time.

b) Give an $O(n^2 \log n)$ time algorithm to pair Ghostbusters with ghosts in such a way that no streams cross.

Q4: **Treaps**

The expected depth of an element in a random treap for a set S is $O(\log n)$, $|S| = n$.

a) Show that the depth of an element in a random treap is $O(\log n)$ with high probability.

b) Show that the height of a random treap is $O(\log n)$ with high probability.

Q5: **Finger searches**

A finger search data structure is a data structure to store a totally ordered set S of elements having a special pointer—the *finger*. The finger always points to the last element accessed in the data structure.

In the following let the data structure storing S by a random skip list where all links are bidirectional, *i.e.*, up and left pointers have been added.

a) Give a skip list implementation of $\mathsf{find}(x)$ that starts the search at the finger and which requires expected time $O(\log d)$, where d is the absolute rank difference in S between x and the element pointed to by the finger. Prove the expected time bound.