Prof. Dr. Susanne Albers
Dr. Suzanne van der Ster
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Institut für Informatik
Technische Universität München

Fall Semester
October 26, 2015

Randomized Algorithms

Exercise Sheet 2

Due: November 2, 2015
at 10:15, in class

Exercise 2.1 (15 points)
To improve the probability of success of the randomized min-cut algorithm, it can be run multiple times.

(a) Consider running the algorithm twice independently. Determine the number of edge contractions and lower bound the probability of finding a min-cut.

(b) Consider the following variation. Starting with a graph with n vertices, first contract the graph down to k vertices using the randomized min-cut algorithm. Make copies of the graph with k vertices, and now run the randomized algorithm on this reduced graph ℓ times, independently. Determine the number of edge contractions and lower bound the probability of finding a minimum cut.

Exercise 2.2 (5 points)
In the min (s,t)-cut problem, we are given an undirected connected multigraph $G = (V,E)$ with two distinguished vertices $s,t \in V$. An (s,t)-cut is a subset of edges $C \subseteq E$ whose removal from G disconnects s from t. The goal is to find an (s,t)-cut of minimum size.
Consider the following adaptation of the min-cut algorithm presented in class for the min (s,t)-cut problem.
The algorithm contracts edges step-by-step. As the algorithm proceeds, the vertices s and t may be replaced by new vertices as a result of edge contractions. However, we ensure that s and t are not in the same vertex at any step of the algorithm.

Find an example of a graph in which the number of distinct min (s,t)-cuts is 2^n.

Exercise 2.3 (10 points)
Let $G = (V, E)$ be an undirected graph. For a set $U \subseteq V$, let

$$\delta(U) = \{uv \in E : u \in U, v \notin U\}$$

be the cut determined by vertex set U.
In this exercise we study the Max Cut problem. In this problem we seek a *maximum cut* for graph G, i.e., we seek the set U that maximizes $\delta(U)$. Unlike the Min Cut problem, this problem is NP-hard.

Consider the following algorithm for approximating the Max Cut problem: Each vertex is added to U independently with probability 1/2. Prove that this gives a cut of size at least $OPT/2$, where OPT denotes the size of the maximum cut in the graph.

Exercise 2.4 (10 points)
Consider the algorithm presented in class to construct a binary planar partition. In the book [MR], the algorithm is also linked to a binary tree representing the partition. In such a tree, every internal node corresponds to a line intersecting a region and every leaf corresponds to a region and the line segment (or portion of it) that it contains. The root corresponds to the entire plane. See Figure 1 for an example for line segments $\{s_1, s_2, s_3\}$.

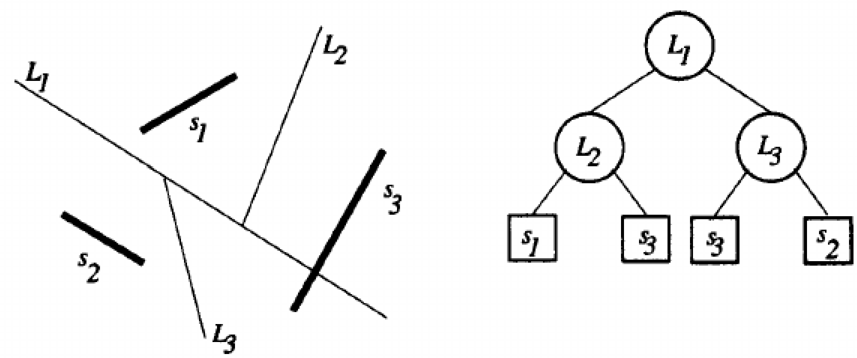

Figure 1: An example of a BPP tree for three line segments.

In class, the following concise description of the **RandAuto** algorithm was given that generates an autopartition.

Algorithm RandAuto
Input: a set $S = \{s_1, \ldots, s_n\}$ *of non-intersecting line segments.*
Output: A binary autopartition P_π *of* S
Pick a permutation π of $\{1, 2, \ldots, n\}$ uniformly at random from the $n!$ possible permutations.
while a region contains more than one segment
    cut it with $l(s_i)$, where i is the first in the ordering π such that s_i cuts that region

Use this as a basis for a elaborate pseudo-code of the algorithm that also outputs the binary planar partition (BPP) tree.

Give the expected running time of your pseudo-code.