Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, A. R. Molla, O. Saukh January 14, 2016

Algorithm Theory, Winter Term 2015/16
Problem Set 11

hand in (hard copy or electronically) by 10:15, Thursday January 21, 2016, tutorial session will be on January 25, 2016

Exercise 1: (7 points)

Assume that we are given a rooted tree with n leaves. All leaves have the same distance h from the root and all nodes except for the leaves have three children. A problem instance consists of such a tree and a boolean value for each leaf.
The value of each inner node is defined as the majority value of its children. Initially, the values of inner nodes are not given. The objective is to compute the value of the root.
The performance of an algorithm to solve this problem is measured by the number of leaves whose values are read by the algorithm.

(a) (2 points) Is there a deterministic algorithm that can solve the problem such that for every input, the algorithm does not need to read the values of all leaves?

(b) (1 points) Design a recursive randomized algorithm to determine the value of the root with certainty but without necessarily reading all the leaves values.

(c) (4 points) What is the expected number of leaves that your randomized algorithm needs to read? Give an upper bound for this expected number of leaves.

Exercise 2: Max-Cut (5 points)

Let $G = (V, E)$ be an undirected graph. Consider the following randomized algorithm: Every node $v \in V$ joins the set S with probaility $1/2$. The algorithm's output is the cut $(S, V \setminus S)$. You can assume that $(S, V \setminus S)$ actually is a cut, i.e., $\emptyset \neq S \neq V$.

(a) (3 points) Show that with probability at least $1/3$ this algorithm outputs a cut which is a 4-approximation to a maximum cut (i.e., a cut of maximum possible size).

Remark: *For a non-negative random variable X, the Markov inequality states that for all $t > 0$ we have $Pr(X \geq t) \leq \frac{E[X]}{t}$.*

Hint: *Apply the Markov inequality to the number of edges* **not** *crossing the cut.*

(b) (2 points) How can you use the above algorithm to devise a 4-approximation of a maximum cut with probability at least $1 - \left(\frac{2}{3}\right)^k$ for $k \in \mathbb{N}$. What is the success probability of your idea.

Remark: *If you could not solve a), you can still use the result as a black box for solving b).*