CS-E3190 Principles of Algorithmic Techniques

09. Tradeoff – Graded Exercise

Please read the following **rules** very carefully.

- Do not consciously search for the solution on the internet.
- You are allowed to discuss the problems with your classmates but you should **write the solutions yourself**.
- Be aware that **if plagiarism is suspected**, you could be asked to have an interview with teaching staff.
- The teaching staff can assist with understanding the problem statements, but will **not be giving any hints** on how to solve the exercises.
- In order to ease grading, we want the solution of each problem and subproblem to start on a **new page**. If this requirement is not met, **points will be deduced**.

1. **Recurrence.** Solve the following recurrence. You can assume that $T(k) = \Theta(1)$ when k is a constant.

$$T(n) = 2T(n/\sqrt{2}) + O(n^2)$$

2. **Induction.** Prove that $p_d \geq \frac{1}{d+1}$ when $p_0 = 1$ and

$$p_d = p_{d-1} - \frac{1}{2}(p_{d-1})^2.$$

   Hint: Function $f(x) = x - \frac{1}{2}x^2$ is increasing for $x \in [0, 1]$.

3. **Improved Algorithm.** We want to devise a faster min-cut algorithm than the Karger's algorithms from the lecture slides and the tutorial exercise. We call this algorithm FASTMINCUT. The high-level idea is to first perform edge contraction, similarly to Karger's algorithm, until we have reduced the number of nodes to t. The probability that we have are left with t nodes such that we have not contracted any edges belonging to the minimum cut is $\binom{t}{2}/\binom{n}{2}$. For $t = n/\sqrt{2}$, this probability is $\approx 1/2$. After this contraction step, we make two recursive calls to FASTMINCUT. Let us define

   - CONTRACT$(G = (V, E), t)$: Run the edge contraction algorithm from the lecture slides until there are t nodes left. The runtime is $O((|V| - t)^2)$.

```
Algorithm 1: FASTMINCUT(G = (V, E))
if |V| < 2√2 then
 | return a min-cut by bruteforce in O(1);
else
 | t ← ⌊|V|/√2⌋;
 | G′ ← CONTRACT(G, t);
 | return min{FASTMINCUT(G′), FASTMINCUT(G′)};
end
```

(a) Compute the success probability of the above algorithm, i.e., with what probability does FASTMINCUT return a minimum cut.

Hint: Observe that the recursion tree is a binary tree of depth d. Apply the result of Problem 2 for said d.

(b) Compute the runtime of FASTMINCUT.

Hint: Use Problem 1.

(c) How many times do we have to run FASTMINCUT in order to boost the probability to $1 - 1/n$? What is the runtime in this case?