CSE202 – FINAL EXAM

The 4 exercises are independent and can be treated in arbitrary order. Within an exercise, the answer to a question can be used in the next ones even if a proof has not been found.

The number of points indicated in front of each question is an indication of the relative difficulties of the questions. It is not necessary to solve all the questions to obtain the maximal possible grade.

Exercise 1. *Let $C(N)$ be the best case for the number of comparisons in an unsuccessful binary seach in a table of length N.*

(1) *(1 pt) Show that $C(N) = 1 + C(\lfloor (N-1)/2 \rfloor)$;*
(2) *(1 pt) Give a formula for largest N such that $C(N) = k$.*

Exercise 2. *Consider a trie built from n random keys over the alphabet $\{0, 1\}$.*

(1) *(2 pts) Show that the average number of nodes examined in an unsuccessful search for an infinite binary string is*

$$\sum_{k \geq 0} (1 - (1 - 2^{-k})^n).$$

(2) *(2 pts) Prove that this number is upper bounded by*

$$\log_2 n + 4.$$

Hint: split the sum into the part for k from 0 to $\lfloor \log_2 n \rfloor$ and its tail, and use the simple bounds

$$e^{-u} \geq 1 - u \quad \text{for } u \in \mathbb{R}, \qquad \log(1-x) \geq -3x/2 \quad \text{for } x \in [0, 1/2].$$

Exercise 3. *Keeping an exact count of a huge number N of events (e.g., packets of data passing through a router) uses $\log N$ bits of memory. Consider the following probabilistic algorithm computing an approximation of N using only $\log \log N$ bits of memory: initialize $k = 0$ and at each new event, increment k with probability 2^{-k}. At the end of the algorithm, return $2^k - 1$. The aim of this exercise is to estimate how good such an algorithm can be.*

Let C_n be the random variable representing the content of the counter after n events and $p_{n,\ell}$ be the probability that $C_n = \ell$.

(1) *(1 pt) Show that*

$$p_{n+1,\ell} = (1 - 2^{-\ell}) p_{n,\ell} + 2^{-(\ell-1)} p_{n,\ell-1}.$$

(2) *(3 pts) Deduce by induction that the expectation of 2^{C_n} is $n+1$, the expectation of 2^{2C_n} is $3n(n+1)/2 + 1$ and the variance of 2^{C_n} is $n(n-1)/2$.*

In order to improve the precision by reducing the variance, consider taking t counters for which the same algorithm is applied in parallel. At the end, the average value $Z_n^{(t)}$ of the estimates $2^{k_1}, \ldots, 2^{k_t}$ is returned.

(3) *(2 pts) Show that there exists a value of t, independent of n, such that*

$$\mathbb{P}\left(\left|\frac{Z_n^{(t)}}{n+1} - 1\right| \geq \frac{1}{4}\right) \leq \frac{1}{4}.$$

In other words, in more than 75% of the executions, the result is within 25% of the exact value.
[Hint: Recall that Chebyshev's inequality states that if a random variable X has finite expectation m and finite non-zero variance σ^2, then for any real $k > 0$, $\mathbb{P}(|X - m| \geq k\sigma) \leq 1/k^2$.]

Exercise 4. *The Knapsack Problem is the problem of packing the maximal value within a bounded weight limit. Formally, it is stated as follows:*

Given positive integers $W, w_1, \ldots, w_\ell$ and $v_1, \ldots, v_\ell$, find a subset $S \subset \{1, \ldots, \ell\}$ that maximizes $\sum_{i \in S} v_i$ subject to $\sum_{i \in S} w_i \leq W$.

Without loss of generality, we assume that $W \geq \max(w_1, \ldots, w_\ell)$.

(1) *(1 pts) Show that the Knapsack Problem is NP-hard.*
(2) *(4 pts) Let $S_{j,v}$ be the smallest weight over subsets of $\{1, \ldots, j\}$ whose total value is v. Show that $S_{j,v} = \min(S_{j-1,v-v_j} + w_j, S_{j-1,v})$. Deduce from there an algorithm that solves the problem in time $O(\ell^2 v_{\max})$, where $v_{\max} = \max(v_1, \ldots, v_\ell)$. Explain why this complexity does not contradict the NP-hardness.*
(3) *(3 pts) Exploit this idea to design a FPTAS for the problem, using values $(\lfloor v_1/K \rfloor, \ldots, \lfloor v_\ell/K \rfloor)$ for a scaling factor K to be determined.*