<div align="center">

Graded Homework 3

</div>

1 Randomized Ski Rental (3 points)

Recall the Ski Rental problem, where you will ski for an unknown number of days, the cost of renting equipments is 1 dollar, and the cost of buying them is B dollars. We saw a simple deterministic 2-competitive strategy in class, which is optimal for deterministic strategies. Here, we see a randomized algorithm that performs better, in expectation.

Consider the following randomized strategy: Randomly select a day $d \leq B$ according to the distribution where $Pr[d = i] \propto (1 - \frac{1}{B})^{B-i}$, with the appropriate normalization. You rent for $d-1$ days and if you reach day d, you buy the equipment. Prove that this strategy has expected competitiveness ratio $\frac{e}{e-1}$.

2 Online Steiner Tree (2 points)

Consider a weighted undirected graph $G = (V, E, w)$. Over time, more and more of the vertices of V become *terminal nodes*. The goal is to find a minimal-cost subtree of G that spans all terminals, i.e., a min-cost Steiner tree. Prove that the following natural online algorithm is $O(\log n)$ competitive: upon a node v becoming a terminal, find the shortest path in G connecting v to the previous terminals, and add all the edges of this shortest path (or a subset of them, when appropriate for maintaining acyclicity) to the tree.

3 Distinct Elements (3 points)

Consider the streaming setting where a stream a_1, a_2, ..., a_m arrives such that each $a_j \in \{1, \ldots, n\}$. We want to estimate the number of distinct elements that appear in the stream, using the following algorithmic idea.

(A) Define a random set S_k by including each $i \in \{1, 2, \ldots, n\}$ in S_k with probability $1/k$. Then, let $y_k = \sum_{i \in S_k} x_i$ where x_i denotes the number of appearances of element $i \in \{1, 2, \ldots, n\}$ in the stream. Prove that whether $y_k = 0$ or not distinguishes whether the number of distinct elements is more than $(1+\varepsilon)k$ or below $(1-\varepsilon)k$, with some probability $\Theta(\varepsilon)$, that is, $Pr[y_k = 0]$ has values that are $\theta(\varepsilon)$ apart, under the two cases that the number of distinct elements is more than $(1 + \varepsilon)k$ or below $(1 - \varepsilon)k$.

(B) Use the above idea to build a Distinct Elements streaming algorithm that outputs an estimate within a $1 \pm \varepsilon$ factor of the actual number of distinct elements with probability at least $1 - \delta$, with space complexity of $O(\log n \cdot \log(1/\delta)/\varepsilon^3)$ bits.

4 Lower Bounds in the Streaming Model (2 points)

Show that a streaming algorithm for Majority Element—working on a stream a_1, a_2, ..., a_m, where each $a_j \in \{1, \ldots, n\}$—that gives the following guarantee needs at least $50 \log n$ bits: the guarantee is that element $i \in \{1, \ldots, n\}$ is certainly output if it appears more than $m/2$ times but it certainly does not get output if it appears less than $m/2(1 - 1/100)$ times.