CS261: Exercise Set #9

For the week of May 25–29, 2015

Instructions:

(1) *Do not turn anything in.*

(2) The course staff is happy to discuss the solutions of these exercises with you in office hours or on Piazza.

(3) While these exercises are certainly not trivial, you should be able to complete them on your own (perhaps after consulting with the course staff or a friend for hints).

Exercise 41

Prove that for every randomized online algorithm $\mathcal{A}$ for the online (integral) bipartite matching problem, there exists a deterministic online algorithm $\mathcal{A}'$ for the online fractional bipartite matching problem such that, for every input and every edge e of the input, the fractional weight that $\mathcal{A}'$ assigns to e equals the probability that $\mathcal{A}$ outputs a matching that includes e.

Conclude that if there is a c-competitive randomized online algorithm for the online (integral) bipartite matching problem, then there is a c-competitive deterministic online algorithm for the online fractional bipartite matching.

Exercise 42

Prove that the converse of Exercise 41 is false.

[Hint: take $\mathcal{A}'$ to be the WaterLevel algorithm from lecture, and consider the 8-cycle.]

Exercise 43

Give an infinite family of instances demonstrating that the greedy algorithm for the online Steiner tree problem is $\Omega(\log n)$-competitive, where n is the number vertices.

[Hint: generalize the example from lecture.]

Exercise 44

Show how to maintain a uniformly random sample from a data stream using a minimal amount of space. More precisely, give a randomized streaming algorithm that keeps track of only a single element $x \in U$ (i.e., $\lceil \log_2 |U| \rceil$ bits) such that, for every data stream $x_1, x_2, \ldots, x_n \in U$, at every time step $i \in \{1, 2, \ldots, n\}$, the current value of x is equally likely to be any of $\{x_1, x_2, \ldots, x_i\}$.

Exercise 45

Prove that every deterministic streaming algorithm that computes the number of distinct elements of a data stream uses $\Omega(n)$ space, where n is the length of the data stream.

[Hint: adapt the reduction from Lecture #18.]