CSC2420: Algorithm Design, Analysis and Theory **Fall 2017**

Lecture 9: Randomized Algorithm

Lecturer: Nisarg Shah *Scribe: Lily Li*

9.1 Sublinear Time Algorithm

Running time in $o(n)$ where n is the length of the input and assuming constant time access to the i^{th} input. Randomization and approximate solution. **Property testing** in accuracy (in detail later on): you are familiar with one and two sized error, here if the property is satisfied you must say YES, if the property is not satisfied but within some ϵ of being satisfied then the algorithm can output anything.

9.1.1 Randomized and Exact

Las Vegas algorithm.

Example 9.1 *INPUT: a sorted doubly linked list L with n elements. Note that this is a sorted* doubly linked list. *Access to the i^{th} entry of the underlying array takes $O(n)$, but if you want the sorted order you have to walk as the linked list (entries of the form $x_i, prev_i, nex_i$). A target x.*

GOAL: determine if $x_i = x$ for some $x_i \in L$.

ALGORITHM: we want an algorithm with expected running time $O(\sqrt{n})$.

9.1.2 Yao's Principle

Claim 9.2 *The expected time of a running time of a randomized algorithm R on the worst input I is no better than the expected time taken under the worst probability distribution D over inputs, by the best deterministic algorithm.*

9.1.3 Randomized and Inexact

Example 9.3 *Estimate average degree in a graph. Let a graph G with $|V| = n$. We have a oracle which can tell us the degree in time $O(1)$.*

9.2 Property Testing

Given some input I, test if I satisfies some property P. If I is satisfying, must output YES. If I is ϵ-far from P then must output NO. If I is ϵ-close to satisfying the property, then we do not care.