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

Lecture 10: Sublinear Time Algorithms

Lecturer: Nisarg Shah *Scribe: Lily Li*

Recall the two axis of consideration: deterministic vs randomized and exact vs inexact. Today we are going to discuss property testing (array is sorted or graph is bipartite), sub-linear space algorithm.

10.1 Sub-linear Time Algorithm Continued

Example 10.1 *We have an array A of length n. Array access takes $O(1)$. We need to check that the array is sorted. Here we assume that the elements of the array are distinct.*

We define ϵ-far to be arrays in which we need to change at least ϵn elements to make A monotonic.

There are many trivial random algorithm that are not good. Consider: (1) randomly pick t elements $a_1, ..., a_t$ of A and check that a_i and its neighbors are sorted. (2) randomly pick t elements $a_1, ..., a_t$ and check that $a_1, ..., a_t$ is a sorted sub-list. Please come up with some counter examples.

Here is an algorithm which works: choose $\frac{2}{\epsilon}$ random indices i in A. Look at the element $A[i]$ and perform binary search for $A[i]$. If we cannot find $A[i]$ then output fail.

Consider the analysis:

Example 10.2 *Property testing on graphs. Since there are two graph representations: adjacency matrix (dense) and adjacency list (list) we consider ϵ-far for each. In the first case we need to change at least ϵn^2 entries to satisfy the property and in the second case we need to change at least ϵnd entries where d is some constant upper bound on the degree (if degree is to high, it makes sense to use adjacency matrices).*

Lets consider bipartite testing in the dense (adjacency matrix) model. The algorithm by Goldreich, Goldwasser, Ron: pick a random subset S of the vertices where $\Theta = \left(\frac{\log(1/\epsilon)}{\epsilon^2}\right)$. Check if the induced graph on S is bipartite. Apparently if the graph is ϵ-far then we reject with probability $\frac{2}{3}$.

Why does this not work as an algorithm for the sparse representation? Look at the definition of ϵ-far for the models. In a way the dense model allows more bad edges. So here is an algorithm for the sparse representation: for $O(1/\epsilon)$ iterations, we will randomly pick a vertex and try to look for odd cycles starting from v. If we do we reject. This had a worse case running time and perform than the algorithm for the dense model, but this is typically the case.

10.2 Sub-linear Space Algorithm

When we talk about sub-linear space, we are talking about sub-linear for the work space. Some unknown fundamental unknown questions:

- $NSPACE(S) = DSPACE(S)$ for $S \geq \log n$?
- $P = L$? (where $L = DSPACE(\log n)$).

- $USTCON$ vs $STCON$ where $USTCON$ is the problem of finding the connectivity of two vertices for an undirected graph.

10.3 Streaming Algorithms

We are given the input data as a stream $a_1, ..., a_m$. Each $a_i \in \{1, ..., n\}$. We wish to use $S(m, n)$ memory where S is sub-linear. It is often the case that m is not known before hand.

Example 10.3 *Consider the missing element problem. (And... we did not talk about the most interesting algorithm.)*

10.3.1 Frequency Moments

Consider next the problem of computing *frequency moments*. Let A be the data stream from above.

10.3.2 Majority Element

You know that the stream is of size m and that there exists some element a_i which appears $\frac{m}{2}$ times. Find this element. This is quite an interesting problem. You should take time and think about it on your own. Bad luck... everyone keeps on talking about the solution. Ok, I know what the algorithm should be.