

6.5240 Sub-linear Time Algorithms

Prof. Ronitt Rubinfeld

TA: Sijin Peng

Course Administrator: Joanne Hanley

What is this course about?

Big data?



Really Big data

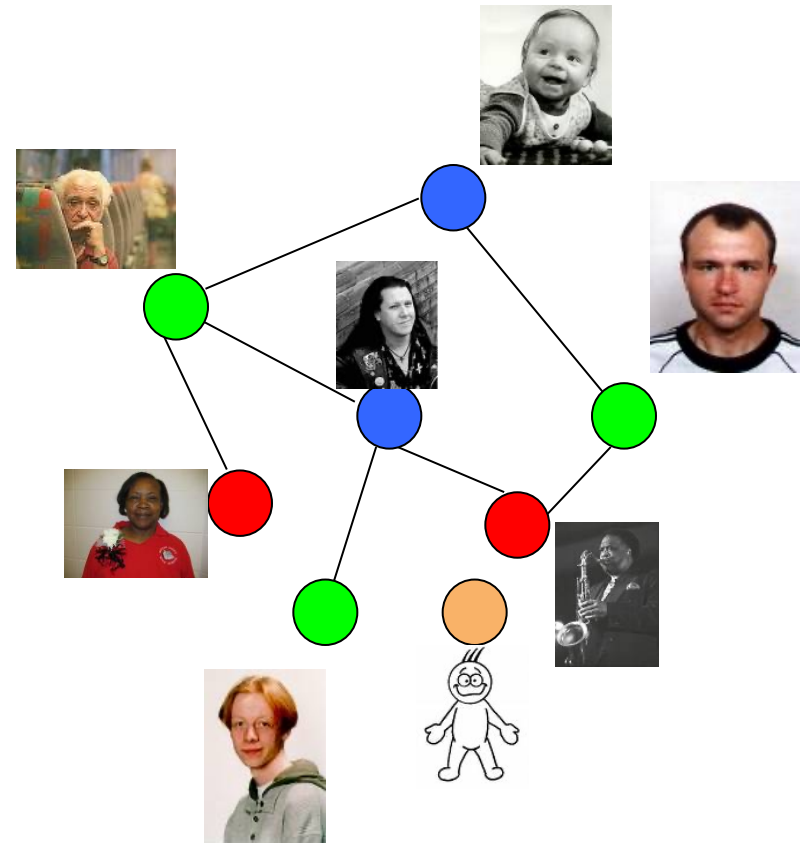
Impossible to access all of it



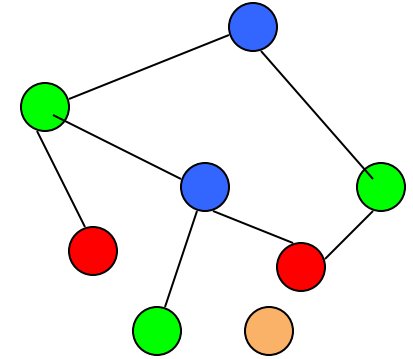
Small world phenomenon

Social network graph:

- each “node” is a person
- “edge” between people that know each other



Connectivity properties



- “*connected*” if every pair can reach each other
- “*distance*” between two nodes is the minimum number of edges to reach one from another
- “*diameter*” is the maximum distance between any pair

Small world property



“Six degrees of separation”

In our language:

diameter of the world population is 6



Does earth have the small world property?

- How can we know?
 - data collection problem is immense
 - unknown groups of people found on earth (e.g., Sentinelese)
 - births/deaths
- Stanley Milgram's 1963 experiment?

The Gold Standard

- linear time algorithms
 - Inadequate?



What can we hope to do without viewing most of the data?

- Can't answer “for all” or “there exists” and other “exactly” type statements:
 - are *all pairs of* individuals connected by at most 6 degrees of separation?
 - *exactly* how many individuals on earth are left-handed?
- Maybe can answer?
 - is there a *large* group of individuals connected by at most 6 degrees of separation?
 - is the *average* pairwise distances of a graph roughly 6?
 - *approximately* how many individuals on earth are left-handed?

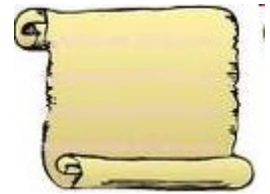
What can we hope to do without viewing most of the data?

- Must compromise:
 - for most interesting problems: algorithm must give *approximate* answer
- we know we can answer *some* questions...
 - e.g., sampling to approximate average, median values

Sublinear time computational models:

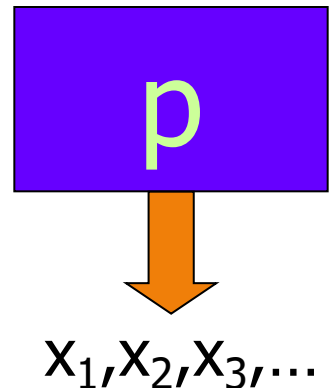
- Random Access Queries

- Can access any word of input in one step
- How is the input represented?



- Samples

- Can get sample of a distribution in one step,
- Alternatively, can only get random word of input in one step
 - When computing functions depending on frequencies of data elements
 - When data in random order



Isn't this just

- Randomized algorithms
- Approximation algorithms
- Statistics
- Learning
- Communication complexity
- Parallel/distributed algorithms?

Course requirements

- Problem sets: 25%
- Midterm (possibly Nov 5): 25%
- Project: 25%
- Scribing, grading and class participation: 25%
 - Scribing:
 - Signup on google doc
 - Must be in latex, using provided style files
 - Draft 2 days after lecture
 - Peer grading

LLM use policy

- Psets: no LLMs (not even grammar, formatting).
- Peer grading: no LLMs
- Scribe notes: no LLMs
- Other uses: ok

Course website

- <https://people.csail.mit.edu/ronitt/COURSE/F26/>
- Announcements
- Pointer to piazza site
- Lecture notes: Posted before lecture
- Psets: Check for updates and hints.
 - Pset 0 is posted! (not to turn in)
- Scribe and grading instructions
- Project ideas
- Probability review

Canvas

- Announcements (with email notification)

Piazza

Please:

help each other without giving too much
information!

be nice to each other!



Caution: anonymous to class but NOT to staff

Project Possibilities

- Read a paper or two or three
 - Explain some lemmas
 - Suggest some open problems
 - Even better -- Make some progress on them, or at least explain what you tried and why it didn't work
- Implement an algorithm or two or three

Can work in groups of 2-3

Plan for this lecture

- Introduce sublinear time algorithms
- Say a bit about the course
- Basic sublinear time algorithms
 - Estimating the diameter
 - Estimating the average degree of a graph



Scribe?

First:

- A very simple example –
 - Deterministic
 - Approximate answer
 - And (of course).... sub-linear time!

Approximate the diameter of a point set

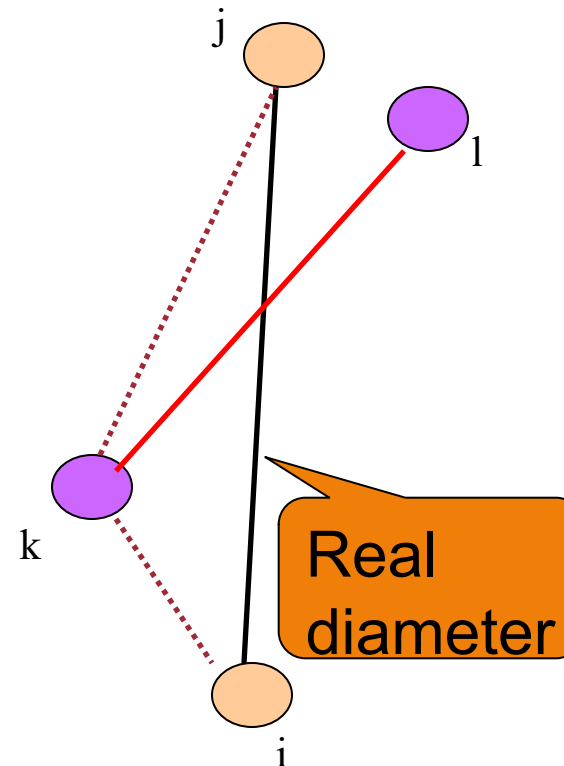
- Given: p points, described by a distance matrix D , s.t.
 - D_{ij} is the distance from i to j .
 - D satisfies **triangle inequality** and **symmetry**.(note: input size $n = p^2$)
- Let i, j be indices that **maximize** D_{ij} then D_{ij} is the **diameter**.
- Output: k, l such that $D_{kl} \geq D_{ij}/2$

2-multiplicative approximation!

Algorithm

- Algorithm:
 - Pick k arbitrarily
 - Pick l to maximize D_{kl}
 - Output D_{kl}
- Running time? $O(p) = O(n^{1/2})$
- Why does it work?

$$\begin{aligned} D_{ij} &\leq D_{ik} + D_{kj} \text{ (triangle inequality)} \\ &\leq D_{kl} + D_{kl} \text{ (choice of } l \text{ + symmetry of } D) \\ &\leq 2D_{kl} \text{ (so } D_{kl} \text{ is at least diameter/2)} \end{aligned}$$



Estimating the average degree

- Given:

- graph $G = (V, E)$

with n vertices m edges,

average degree $\bar{d} \equiv \frac{1}{n} \cdot \sum_{u \in V} d(u) = 2m/n$

- Approximation parameter ϵ

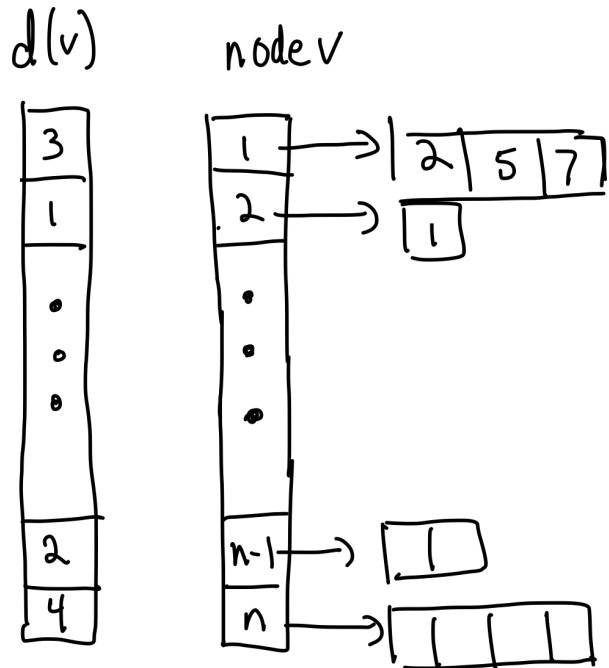
- Confidence parameter δ

e.g. 1/4

- Goal:

- Output $\tilde{d}$ such that $\Pr[|\tilde{d} - \bar{d}| \leq \epsilon \cdot \bar{d}] \geq 1 - \delta$

Access to graph?



- Neighbor queries (adjacency list):
 - Given (v, j) output j^{th} neighbor of v
- Degree queries:
 - Given v output degree of v : $d(v)$

A first idea: Naïve sampling

Algorithm:

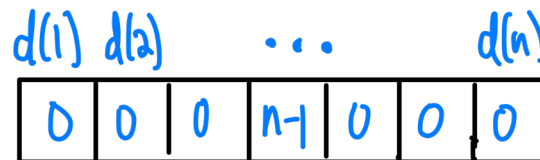
- Pick $O(??)$ sample nodes $v_1, \dots, v_s$
- Output average degree of sample:

$$\frac{1}{s} \cdot \sum_i d(v_i)$$

How many samples?

Straightforward Chernoff/Hoeffding bounds $\Omega(n)$

Lower bound?



need $\Omega(n)$ samples to
find "needle in haystack"

A first idea: Naïve sampling

Algorithm:

- Pick $O(??)$ sample nodes $v_1, \dots, v_s$
- Output average degree of sample:

$$\frac{1}{s} \cdot \sum_i d(v_i)$$

Lower bound?

$d(1)$	$d(2)$		$\dots$		$d(n)$
0	0	0	$n-1$	0	0

need $\Omega(n)$ samples to
find "needle in haystack"

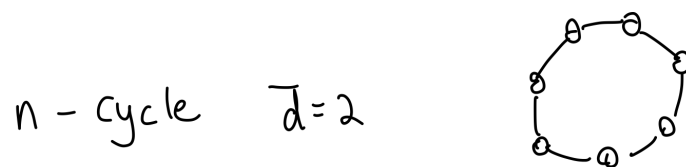
Not a possible degree sequence!!

$n-1$	1	1	1	1	1	1
-------	---	---	---	---	---	---

Is possible!

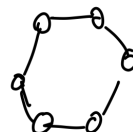
Some lower bounds:

- “Ultrasparse” case:
 - 0 edges vs. 1 edge
 - Need $\Omega(n)$ queries to distinguish
 - Yields lower bound on multiplicative approximation
- Average degree 2 example:



vs.

$n \cdot c \cdot \sqrt{n}$ cycle $\bar{d} \approx 2 + c^2$
+ $c\sqrt{n}$ -clique



Need $\Omega(\sqrt{n})$ queries
to find clique node

Assumptions

1. Average degree $\bar{d} > 1$
2. G is simple

Warmup 0: Regular graphs

Assumption: each node has degree Δ

Algorithm:

- Output Δ

