DS-563 / CS-543 Lecture 10 / 2023-02-23

Main topic today: graph connectivity sketches

Problem:

Input: a stream describing a graph G on $V = [n]$

Question: is G connected?

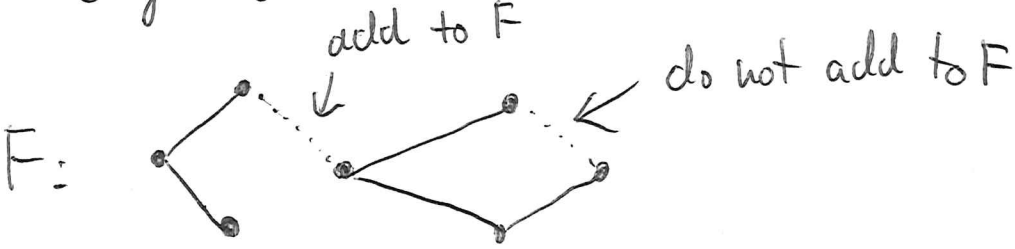
Warm-up: Insertion-only stream

Input: $(2,3), (1,4), (7,2), \ldots$

Solution: - keep a set of edges $= F$ that span the graph seen so far

- no cycles, so it's a forest

New edge arrives:



$O(n)$ words of space: we store $\leq n-1$ edges

Fast version: use the union-find data structure

Insertion-deletion streams

Input: "insert $(2,3)$", "insert $(1,3)$", "remove $(2,3)$", ...

[discussion: how to attack this problem]

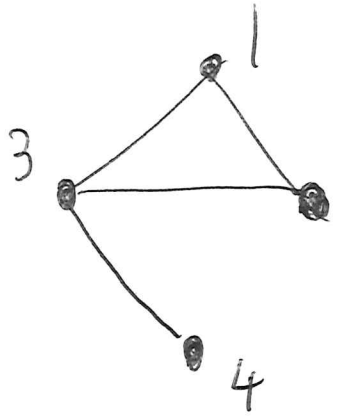
Ingredient 1 (out of 3): graph encoding

$G \Rightarrow$ [matrix] ← row for each vertex

↑ column for each potential edge

n rows $\times \binom{n}{2}$ columns

Example:



	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$x_1 =$	1	1	0	0	0	0
$x_2 =$	-1	0	0	1	0	0
$x_3 =$	0	-1	0	-1	0	1
$x_4 =$	0	0	0	0	0	-1

Entry corresponding to vertex i and edge $\{a,b\}$: $(a<b)$

if $i \neq a$ and $i \neq b$: 0

if $\{a,b\}$ not in G : 0

if $\{a,b\}$ in G and $i = a$: 1

if $\{a,b\}$ in G and $i = b$: -1

Observation: non-zero entries of x_i correspond to edges connecting i to other vertices

Definition: for $S \subseteq V$, $x_S = \sum_{i \in S} x_i$

Claim: non-zero entries of x_S correspond to edges connecting a vertex in S to a vertex in $V \setminus S$

Proof sketch: $e = \{a,b\}$ ← arbitrary edge

Case 1: e not in the graph

all $(x_i)_e = 0$ so $(x_S)_e = 0$ ✓

Case 2: e in G, $a, b \in S$

Two non-zero entries in column e, in rows a, b.

$(x_S)_e = (x_a)_e + (x_b)_e = (+1) + (-1) = 0$ ✓

10-3

Case 3: e in G, $a, b \in V \setminus S$

Non-zero entries in column e, not in the sum for x_S. So $(x_S)_e = 0$ ✓

Case 4: e in G, one of a/b in S, one in $V \setminus S$

Exactly one ~~entry for~~ non-zero entry in column e included in sum x_S.

So $(x_S)_e \in \{-1, 1\}$. ✓

Ingredient 2: Borůvka's algorithm

for each vertex:
- create component that contains this vertex

repeat:
- for each component C:
  - select arbitrary edge connecting C to the rest of the graph (if possible)
- merge components connected via selected edges

Claim: Before i-th iteration, each component either maximal or its size $\geq 2^{i-1}$

Proof sketch: induction

$i=1$: size of each component $\geq 2^{1-1} = 2^0 = 1$ ✓

$i > 1$: assume true for ~~smaller i~~ $0, 1, \ldots, i-1$

C - arbitrary component

if maximal, done ✓

if not maximal, was merged from 2 or more non-maximal components,

their size was $\geq 2^{(i-1)-1} = 2^{i-2}$

after merging size of $C \geq 2 \cdot 2^{i-2}$

$= 2^{i-1}$

✓

Corollary: $\lceil \log n \rceil$ iterations ~~rounds~~ suffice to discover all connected components

Ingredient 3: L_0-sampling

There is a linear sketching algorithm that takes a vector in $\{-n, \ldots, n\}^n$ and turns it into polylog(n) bits s.t.

- the algorithm operates in polylog(n) space.
- for any vector v, the algorithm correctly reports a non-zero coordinate of v or reports that v is all zeros, with probability $1 - \frac{1}{n^3}$.

Note: "sampling" means report uniformly at random from non-zero coordinates, not needed today!

[if time allows, discuss some ideas how to implement this]

Combining ingredients

Main ideas:

- simulate Borůvka for $\lceil \log n \rceil$ iterations
- use the graph encoding, but store its l_0-sampling sketch for each vertex

For i-th iteration of Borůvka:

- For each current component C:
  - compute l_0-sampling sketch for x_C by adding sketches for all $i \in C$
  - use l_0-sampling to find a non-zero coordinate in x_C and corresponding edge (if possible)
- Merge components connected by discovered edges

Notes:

- use fresh l_0-sampling sketches for each iteration
  - adaptivity could be a problem!
- Space usage:

$$\underbrace{\lceil \log n \rceil}_{\#\text{iterations}} * \underbrace{n}_{\#\text{vertices}} * \underbrace{\text{polylog}(n)\text{ bits}}_{l_0\text{-sampling sketch of graph encoding's row for single vertex}} = O(n \,\text{polylg}(n))$$

Probability of failure: via union bound

$$\leq \underbrace{\frac{1}{n^3}}_{\text{Probability a single } l_0\text{-sampling fails}} \cdot \underbrace{\lceil \log n \rceil \cdot n}_{\#\text{ of applications to consider}} = O\left(\frac{1}{n}\right) = \text{small}$$