Finding Matchings in the Semi-Streaming Model

Kheeran Naidu

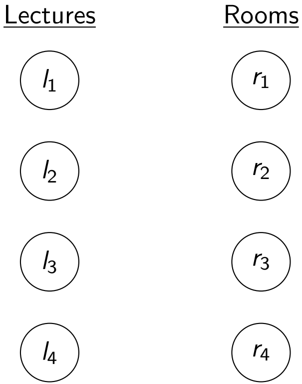
kn16063@bristol.ac.uk

Joint work with Dr Christian Konrad

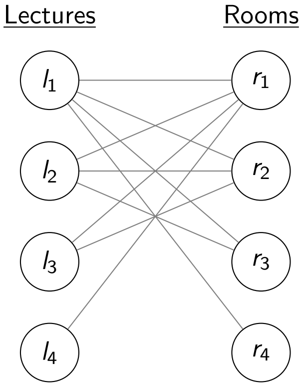
Overview

Overview

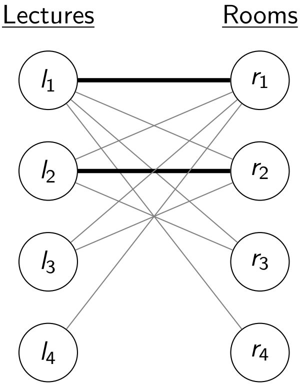
Introduction



Introduction

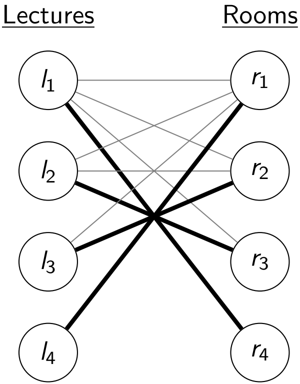


Introduction



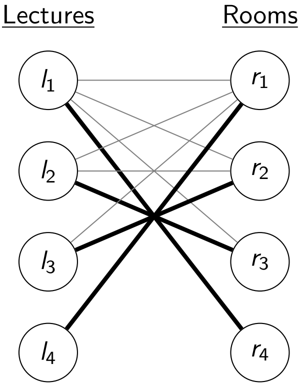
Maximal Matching

Introduction



Maximum Matching

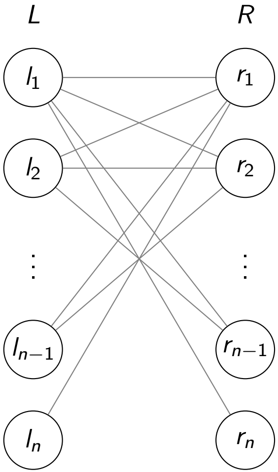
Introduction



Bipartite Graph $G = (L, R, E)$:

- **Maximal** & **Maximum** Matchings

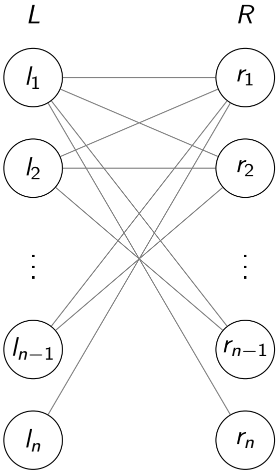
Introduction



Bipartite Graph $G = (L, R, E)$:

- **Maximal** & **Maximum** Matchings
- $|L| = |R| = n$ **vertices**
- $|E| \leq n^2$ **edges**, $O(n^2)$

Introduction



Bipartite Graph $G = (L, R, E)$:

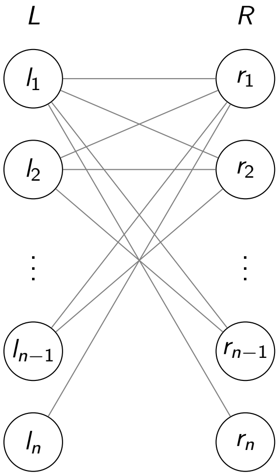
- **Maximal** & **Maximum** Matchings
- $|L| = |R| = n$ **vertices**
- $|E| \leq n^2$ **edges**, $O(n^2)$

Other examples: ride hailing, online advertising, etc.

Classic Algorithms

Finding maximum matchings:

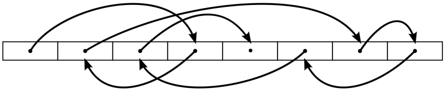
- Edmonds-Karp algorithm
- Dinitz's algorithm
- Hopcroft-Karp algorithm



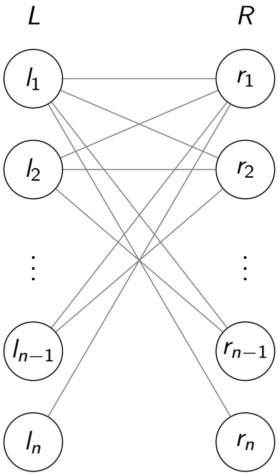
Finding maximum matchings:

- Edmonds-Karp algorithm
- Dinitz's algorithm
- Hopcroft-Karp algorithm

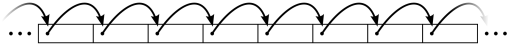
Limitation: assumes **random access** to the edges of the graph



Random Access Memory (RAM) is expensive. Typically 16GB of RAM in a computer.



Semi-Streaming Model

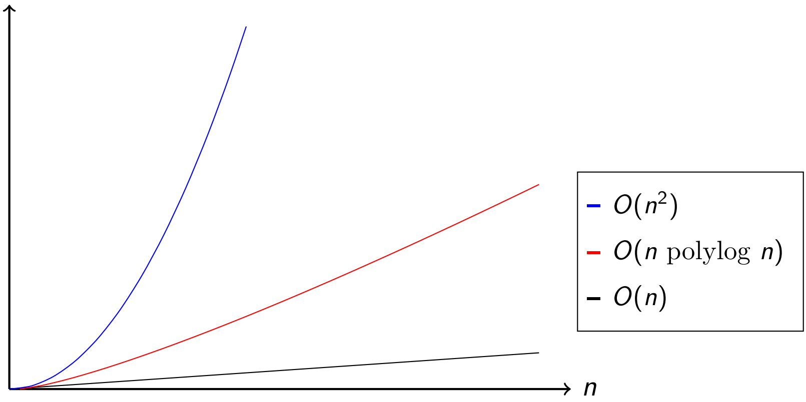


Feigenbaum et al. [ICALP04]:

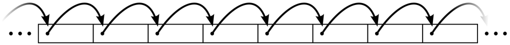
- Only allows **sequential access** to the edges of the graph.
- Algorithms with memory $O(n \operatorname{polylog} n) = O(n\ (\log n)^c)$.

Semi-Streaming Model

Growth Rate of Functions



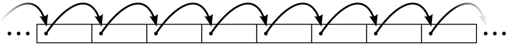
Semi-Streaming Model



Feigenbaum et al. [ICALP04]:

- Only allows **sequential access** to the edges of the graph.
- Algorithms with memory $O(n \operatorname{polylog} n) = O(n\ (\log n)^c)$.

Semi-Streaming Model

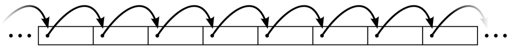


Feigenbaum et al. [ICALP04]:

- Only allows **sequential access** to the edges of the graph.
- Algorithms with memory $O(n \operatorname{polylog} n) = O(n\ (\log n)^c)$.

Example: For a dense graph taking 1TB of memory, a semi-streaming algorithm is allowed about 1GB of memory.

Semi-Streaming Model



Feigenbaum et al. [ICALP04]:

- Only allows **sequential access** to the edges of the graph.
- Algorithms with memory $O(n \operatorname{polylog} n) = O(n (\log n)^c)$.
- Ideally with few passes of the sequence of edges (stream).

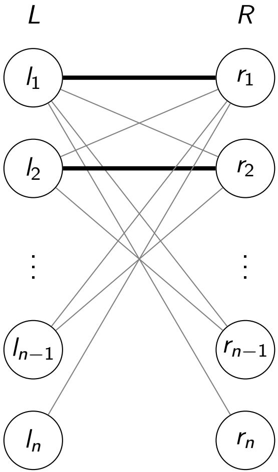
Example: For a dense graph taking 1TB of memory, a semi-streaming algorithm is allowed about 1GB of memory.

Overview

One-Pass Algorithm

A simple GREEDY algorithm is the best known algorithm:

- finds a maximal matching
- is always at least $0.5\times$ the size of the maximum matching
- 0.5-approximation

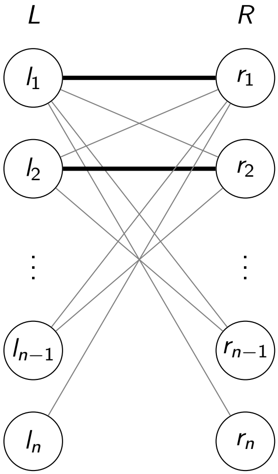


General Approach

Two-Pass Algorithm

Class of algorithms:

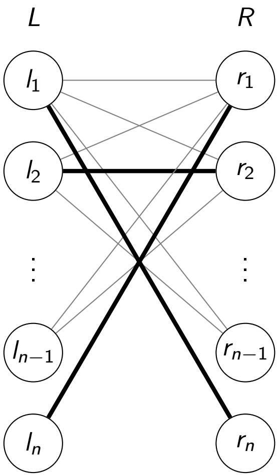
1. (first pass) finds a maximal matching M using GREEDY;
2. (second pass) increases the size of the matching M.



Two-Pass Algorithm

Class of algorithms:

1. (first pass) finds a maximal matching M using GREEDY;
2. (second pass) increases the size of the matching M.



General Approach

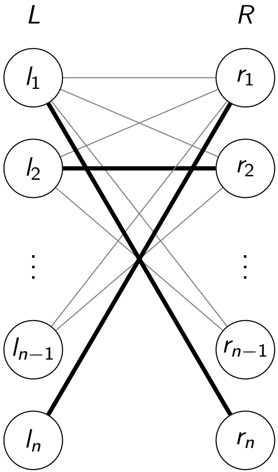
Two-Pass Algorithm

Class of algorithms:

1. (first pass) finds a maximal matching M using GREEDY;
2. (second pass) increases the size of the matching M.

Our Work

Algorithmic: 0.585-approximation
Impossibility: 0.667-approximation



Summary

	Algorithmic	Impossibility
one-pass	0.5 [folklore] (GREEDY)	0.667 [SODA12] 0.632 [SODA13] 0.591 [SODA21]
two-pass	0.519 [APPROX12] 0.563 [APPROX17] 0.583 [ICDMW16] 0.585 [MFCS18] $2 - \sqrt{2} \approx 0.585$	$\frac{2}{3} \approx 0.667$[1]

[1] where the first pass runs GREEDY, i.e., finds at least a 0.5-approximation.

Thank You