CSCE 689: Special Topics in Modern Algorithms for Data Science

Lecture 19

Samson Zhou

Presentation Schedule

- November 27: Chunkai, Jung, Galaxy AI
- November 29: STMI, Anmol, Jason
- December 1: Bokun, Ayesha, Dawei, Lipai

Previously in the Streaming Model

- Reservoir sampling
- Heavy-hitters
  - Misra-Gries
  - CountMin
  - CountSketch
- Moment estimation
  - AMS algorithm
- Sparse recovery
- Distinct elements estimation

Reservoir Sampling

- Suppose we see a stream of elements from $[n]$. How do we uniformly sample one of the positions of the stream?

47 72 81 10 14 33 51 29 54 9 36 46 10

Heavy-Hitters (Frequent Items)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let L_p be the norm of the frequency vector:

$$L_p = \left(f_1^p + f_2^p + \cdots + f_n^p\right)^{1/p}$$

- Goal: Given a set S of m elements from $[n]$ and a threshold ε, output the elements i such that $f_i > \varepsilon\, L_p$...and no elements j such that $f_j < \frac{\varepsilon}{2}\, L_p$ (we saw algorithms for $p = 1$ and $p = 2$)
- Motivation: DDoS prevention, iceberg queries

Frequency Moments (L_p Norm)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let F_p be the frequency moment of the vector:

$$F_p = f_1^p + f_2^p + \cdots + f_n^p$$

- Goal: Given a set S of m elements from $[n]$ and an accuracy parameter ε, output a $(1+\varepsilon)$-approximation to F_p
- Motivation: Entropy estimation, linear regression

Distinct Elements (F_0 Estimation)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let F_0 be the frequency moment of the vector:

$$F_0 = |\{i : f_i \neq 0\}|$$

- Goal: Given a set S of m elements from $[n]$ and an accuracy parameter ε, output a $(1 + \varepsilon)$-approximation to F_0
- Motivation: Traffic monitoring

Sparse Recovery

- Suppose we have an insertion-deletion stream of length $m = \Theta(n)$ and at the end we are promised there are at most k nonzero coordinates
- Goal: Recover the k nonzero coordinates and their frequencies

The Streaming Model

- So far, all questions have been *statistical*

- What other questions can be asked? (Think in general, outside of the streaming model)

The Streaming Model

- So far, all questions have been *statistical*

- What other questions can be asked? (Think in general, outside of the streaming model)

- Algebraic, geometric

The Streaming Model

- So far, all questions have been *statistical*

- What other questions can be asked? (Think in general, outside of the streaming model)

- Algebraic, geometric

TODAY

Graph Theory

- Suppose we have a graph G with vertex set V and edge set E

- Let $V = [n]$ for simplicity, so each vertex is an integer from 1 to n

- Then each edge $e \in E$ can be written as $e = (u, v)$ for $u, v \in [n]$
- In other words, each edge is a pair of integers from 1 to n

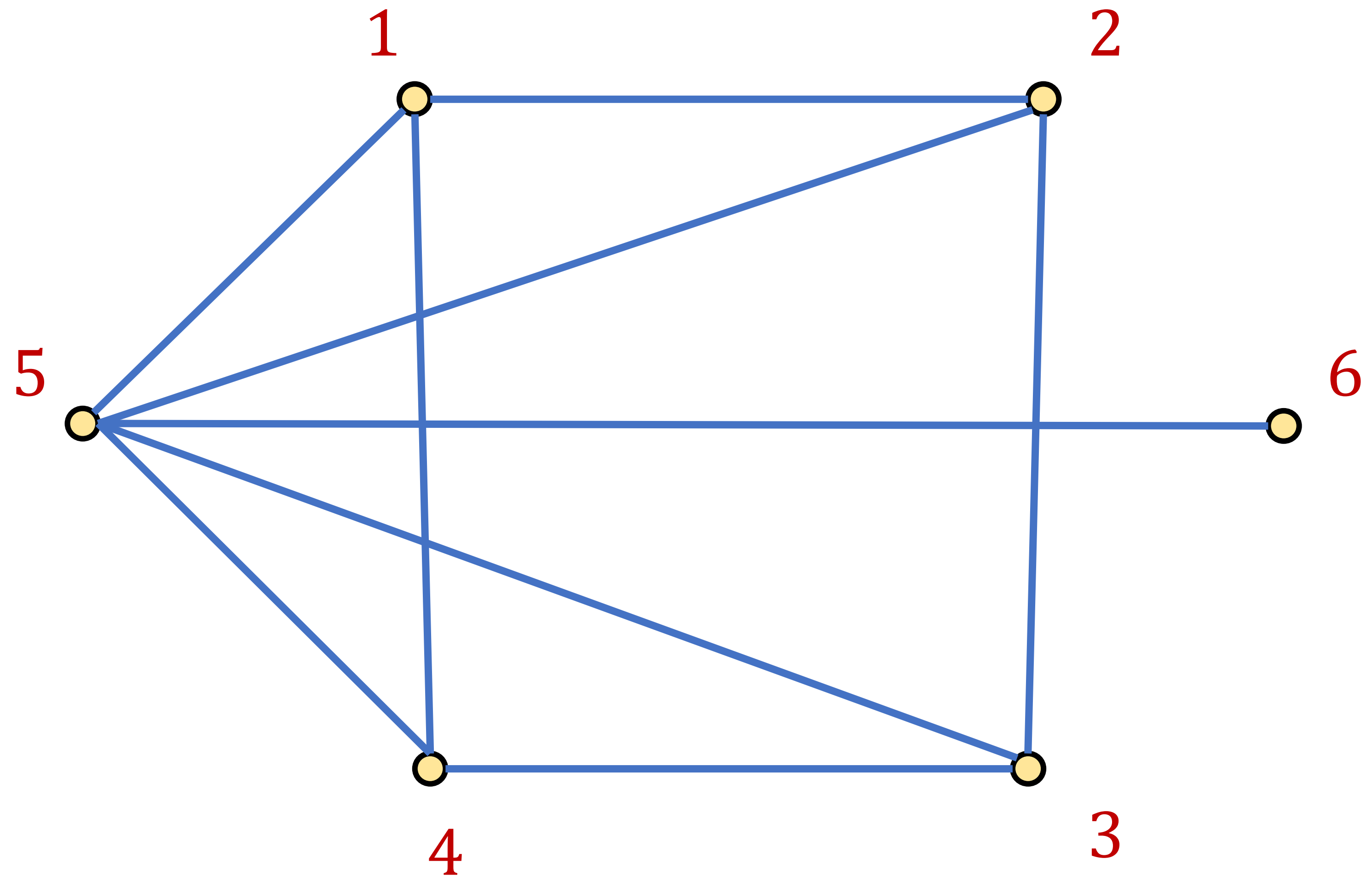
Graph Theory

- For today, we will assume a simple, undirected, unweighted graph

- Graph has no self-loops, no multi-edges

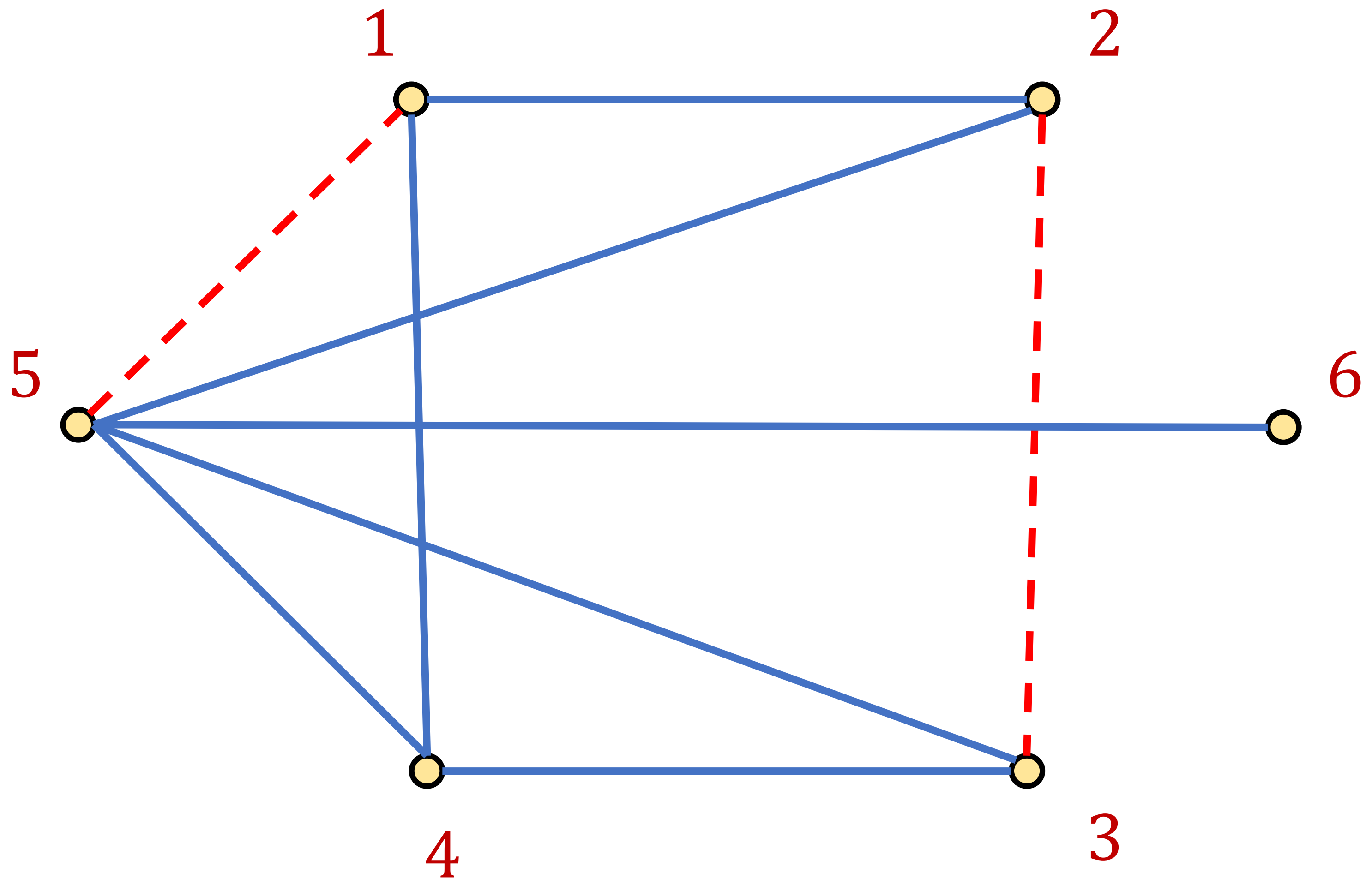
- Edges are undirected

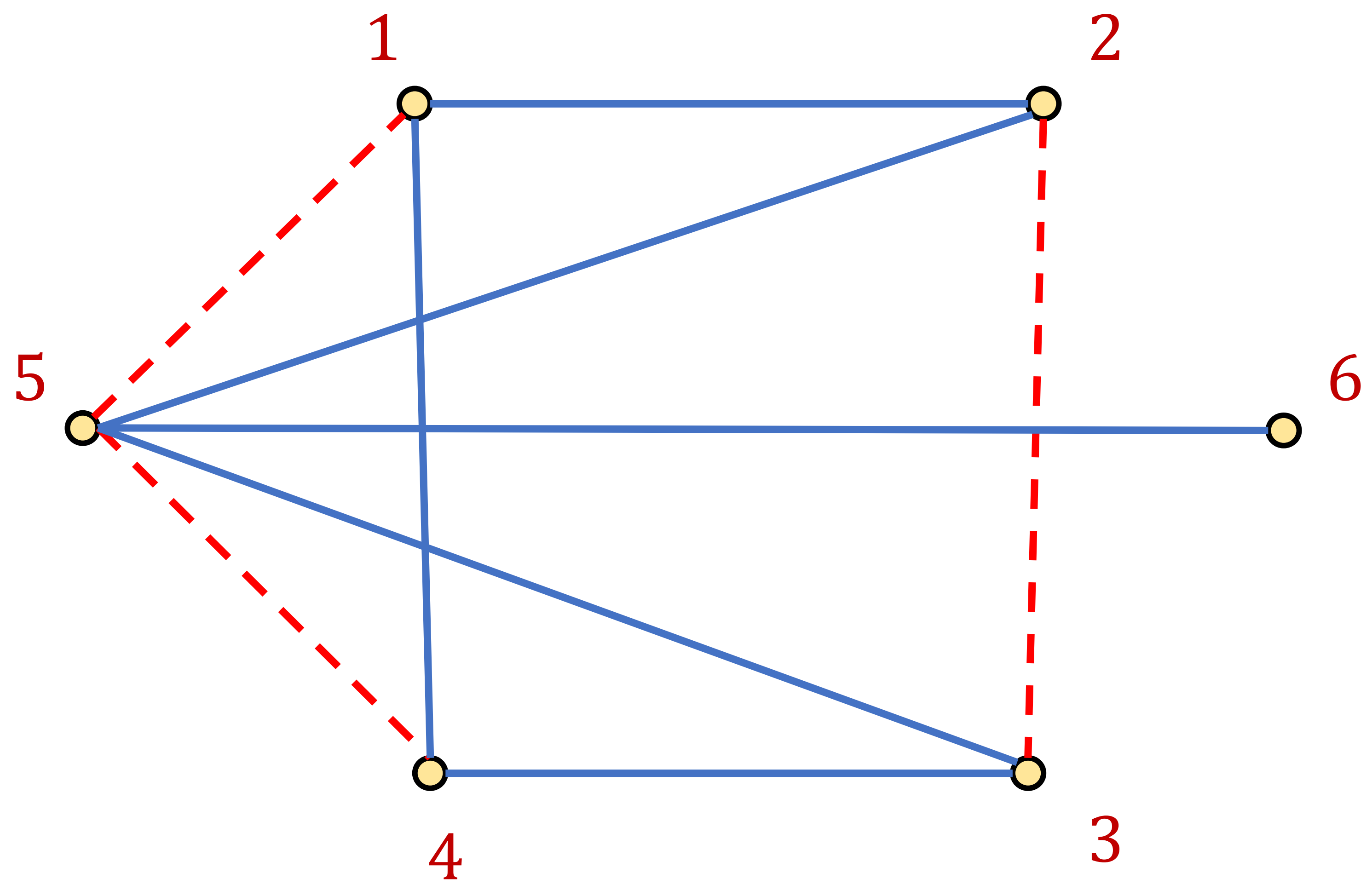
- Each edge has weight 1


1
2
5
6
4
3

Matchings

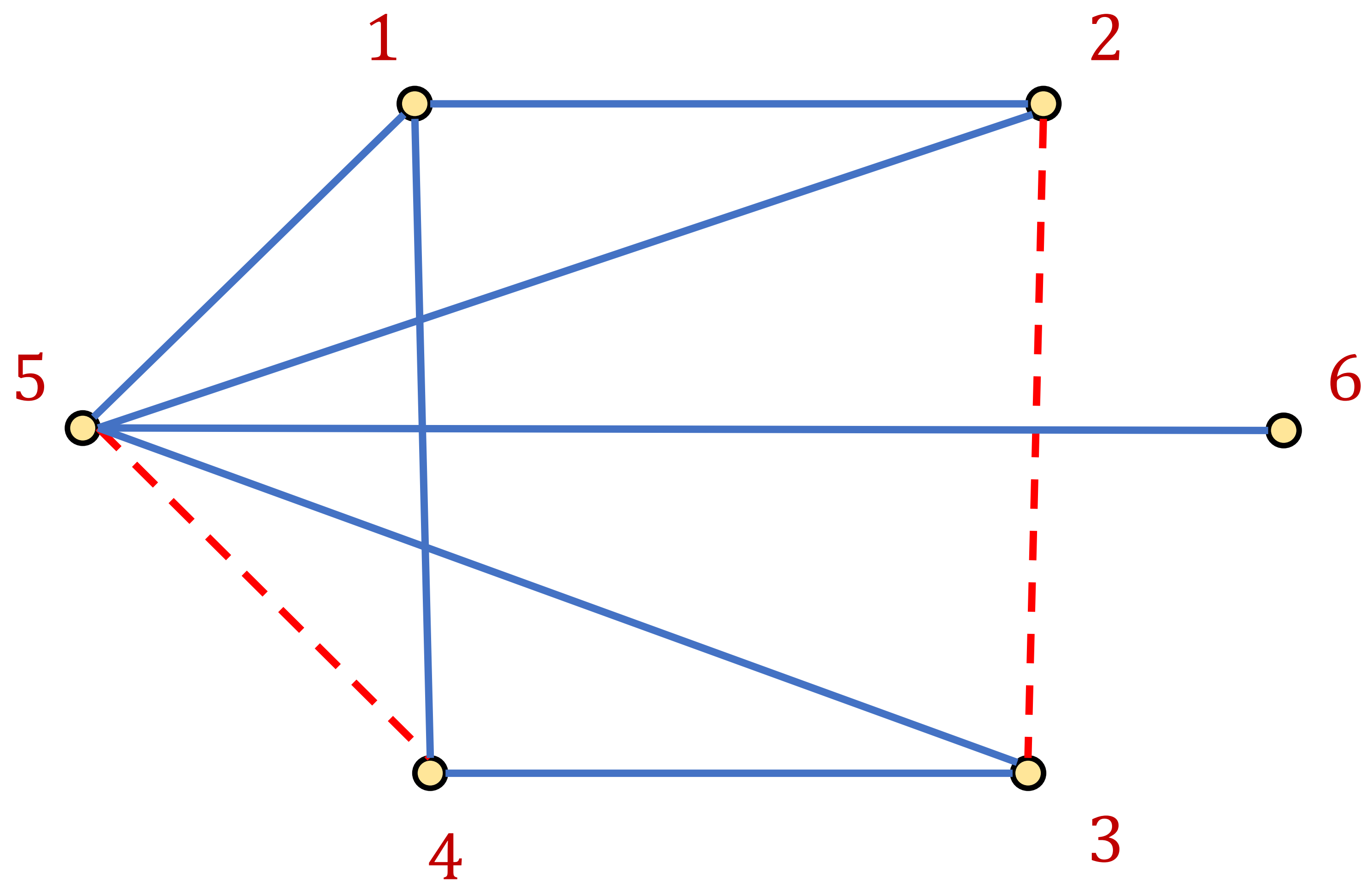
- A matching M is a subset of edges of E such that no two edges share a common vertex


1
2
5
6
4
3


1
2
5
6
4
3

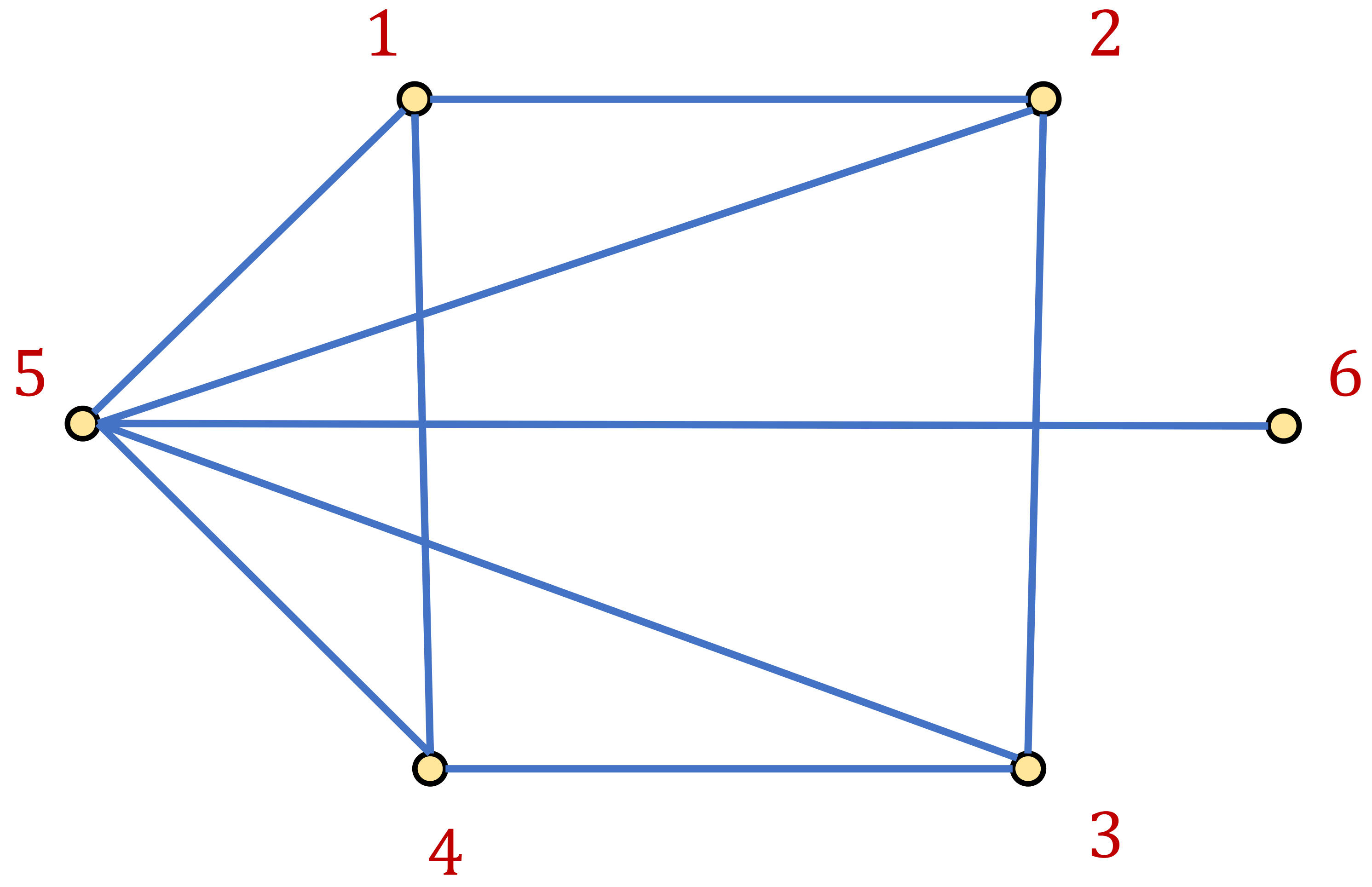
Maximal Matching

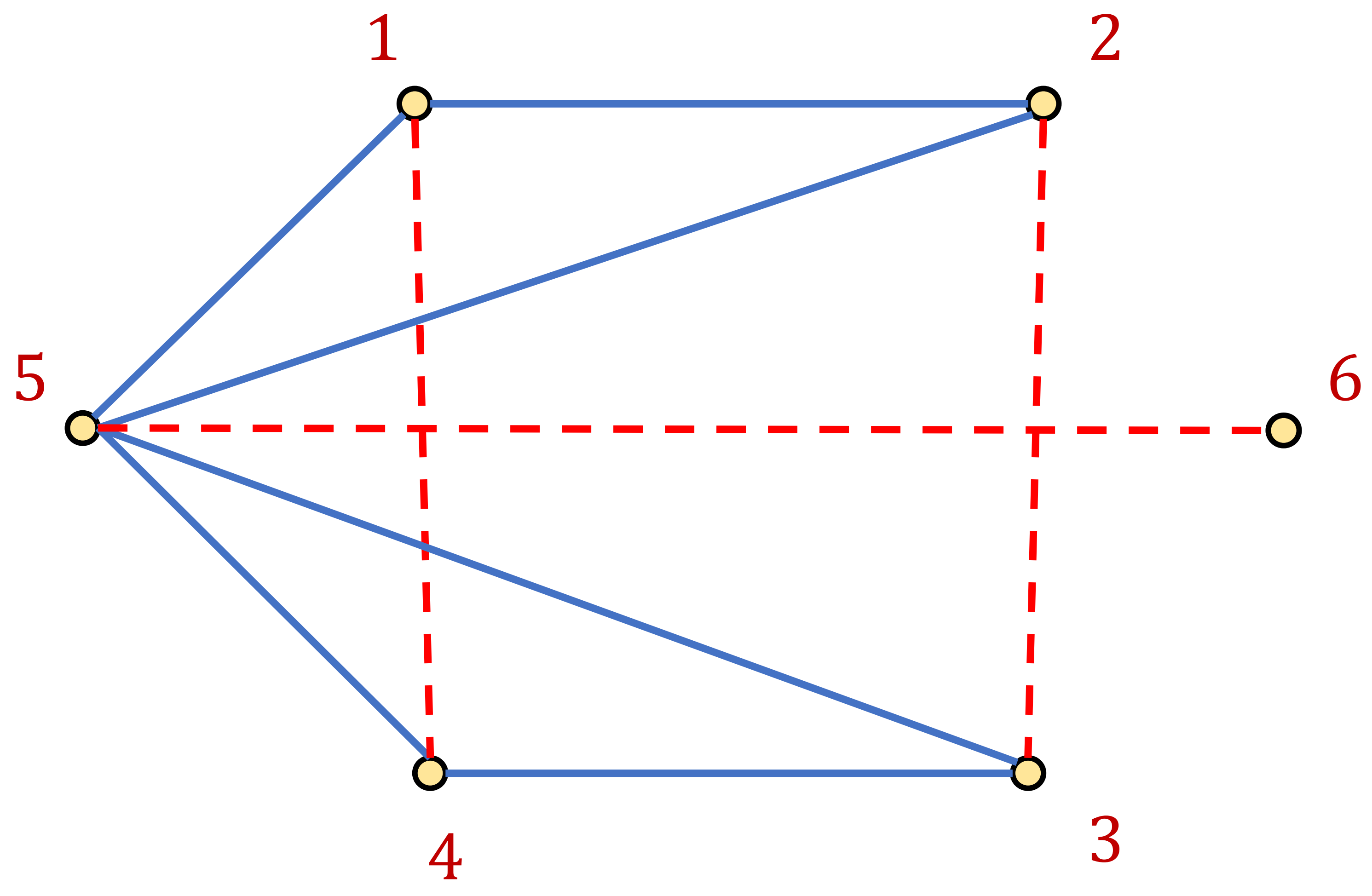
- A maximal matching M of G such that any additional edges would no longer be a matching


1
2
5
6
4
3

Maximum Matching

- Find a matching M of G with the largest possible number of edges


1
2
5
6
4
3

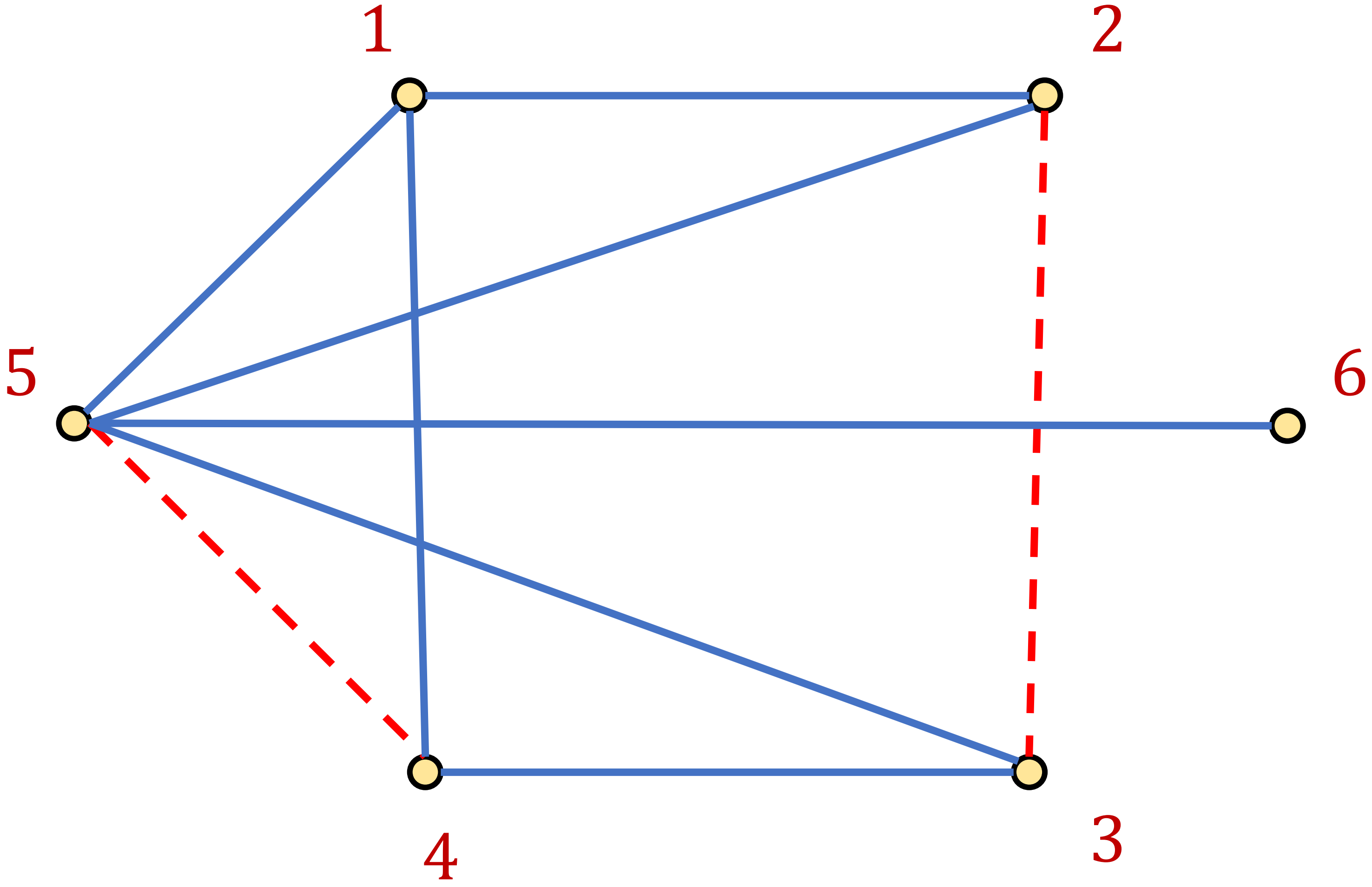

1
2
5
6
4
3

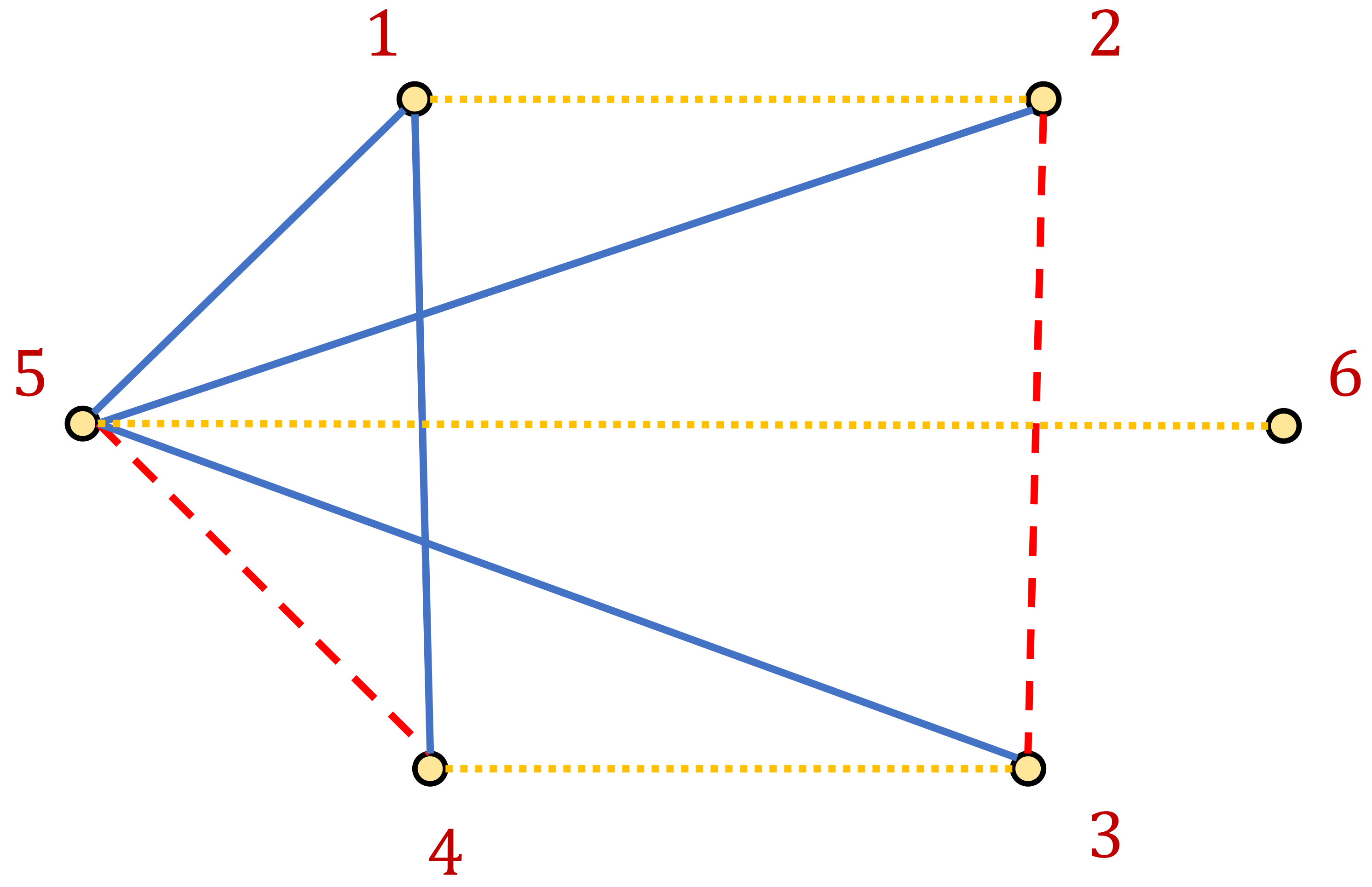
Applications for Maximum Matching

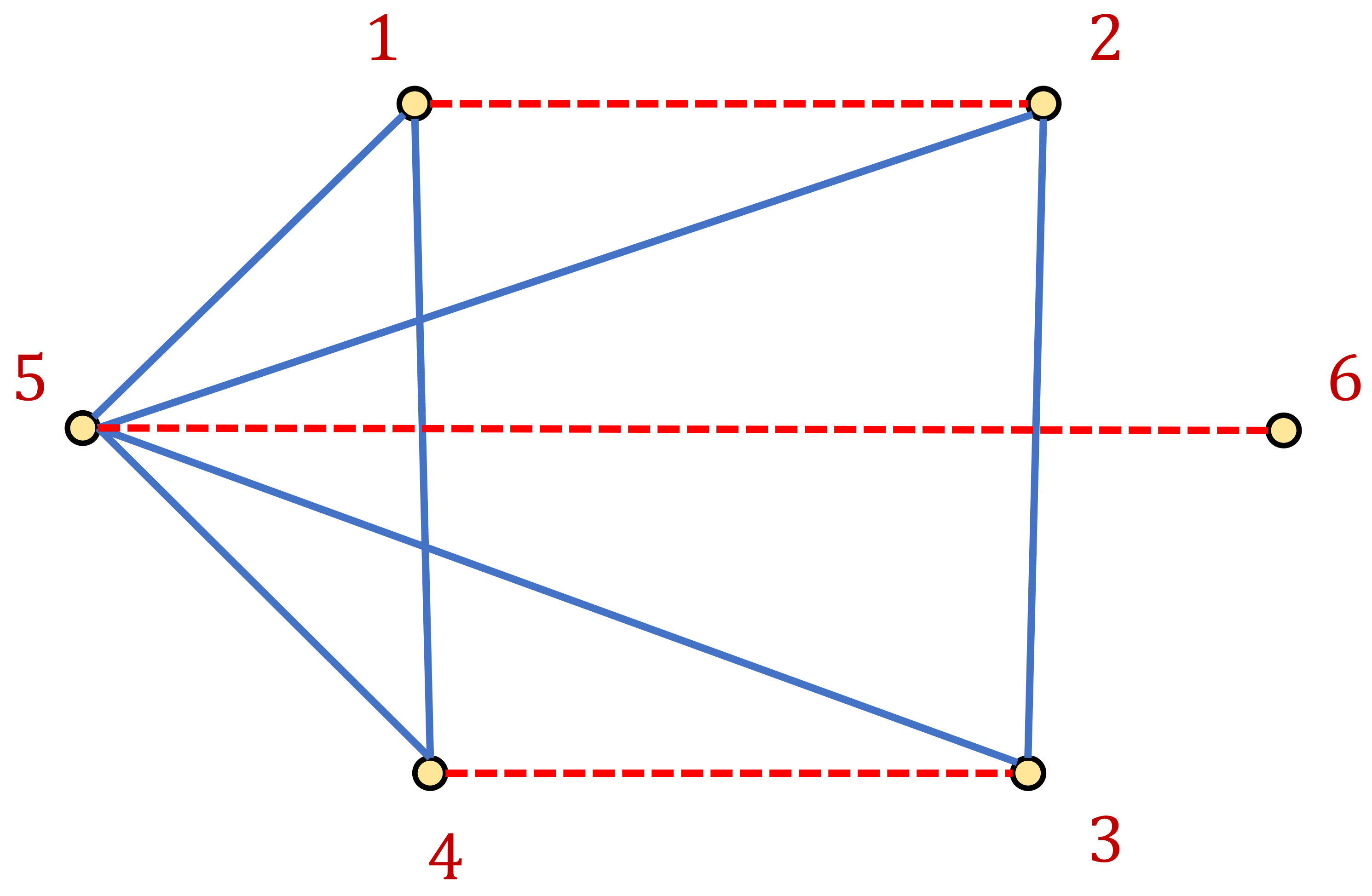
- Fill the largest number of positions with applicants across a system

Maximum Matching

- How to find maximum matching?
- An *alternating path* is any path of edges that alternates between edges in and not in the matching
- An *augmenting path* is any alternating path of edges that does not start and does not end at a vertex in the matching
- "Flipping" all the edges in an augmenting path increases the matching size


1
2
5
6
4
3


1
2
5
6
4
3


1
2
5
6
4
3

Maximum Matching

- It turns out repeatedly finding *augmenting paths* is sufficient for finding a maximum matching

- Formally: If a matching is not a maximum matching, there exists an augmenting path to the matching

- Algorithms by Hopcroft and Karp (1973) and Edmonds (1965) for finding augmenting paths – can be done in polynomial time

Semi-streaming Model

- Recall that we have a graph $G = (V = [n],\ E)$
- Suppose $|E| = m$
- The edges of the graph arrive sequentially, i.e., insertion-only model
- We are allowed to use $n \cdot \mathrm{polylog}(n)$ space
- Enough to store a matching, NOT enough to store entire graph, since m can be as large as $O(n^2)$

Semi-streaming Model

- Can we run the augmenting paths algorithm?

Semi-streaming Model

- Can we run the augmenting paths algorithm? Not clear…

- In fact, Kapralov (2013) showed NO one-pass semi-streaming algorithm for maximum matching can achieve approximation better than $\frac{e}{e-1} \approx 1.582$

Maximal Matching

- What if we just wanted to find a maximal matching?

Maximal Matching

- What if we just wanted to find a maximal matching?

- Greedy algorithm: Add each unmatched edge e in the stream to the matching M