CSCE 658: Randomized Algorithms

Lecture 2

Samson Zhou

Last Time: Schwartz-Zippel Lemma

- [Schwartz-Zippel] Suppose P is a degree d polynomial in $x_1, \dots, x_n$. Let $r_1, \dots, r_n$ be randomly drawn from $\{1,2,3, \dots, q\}$. Then

$$\Pr[P(r_1, \dots, r_n) = 0] \leq \frac{d}{q}$$

- Upshot: A random evaluation of a low-degree polynomial is unlikely to be zero

Last Time: Equality Problem

- Alice is given a string A and Bob is given a string B, each of length n, and they must determine whether $A = B$, using the *minimum amount of communication*

- Any deterministic protocol must use $\Omega(n)$ bits of communication, but there exists a randomized protocol that uses $O(\log n)$ bits of communication

Last Time: Equality Problem

- Algorithm: Suppose Alice and Bob have access to a randomly generated string $x \in \{1,2,3,\ldots,q\}^n$. Alice sends over Ax and Bob determines whether $Ax = Bx$

- If $A = B$, then $Ax = Bx$ so the protocol succeeds
- If $A \neq B$, then what is the probability that $Ax \neq Bx$?
- By Schwartz-Zippel, the probability that $Ax \neq Bx$ is at least $\frac{9}{10}$

Polynomial Identity Testing

- $f(x,y) = x^2 - y^2$
- $g(x,y) = (x+y)(x-y)$

- Do we have $f(x,y) \equiv g(x,y)$?

Polynomial Identity Testing

- $f(x, y) = x^3 + 3xy + y^3 - 1$
- $g(x, y) = \frac{1}{2}(x + y - 1)\left((x + 1)^2 + (y + 1)^2 + (x - y)^2\right)$

- Do we have $f(x, y) \equiv g(x, y)$?

Polynomial Identity Testing

- $f(x, y) = x^3 + 3xy + y^3 - 1$
- $g(x, y) = \frac{1}{2}(x + y - 1)\left((x + 1)^2 + (y + 1)^2 + (x - y)^2\right)$

- Do we have $f(x, y) \equiv g(x, y)$?

- Both are equal to $h(x, y) = (x + y - 1)(x^2 - xy + y^2 + x + y + 1)$

Polynomial Identity Testing

- Efficiently determine whether polynomials of degree d satisfy $f(x_1, \ldots, x_n) \equiv g(x_1, \ldots, x_n)$

- Why not just expand the polynomials and see whether they are equal?

- How many terms can be in $(x_1 + x_2 + \cdots + x_n)^d$?

Polynomial Identity Testing

- Efficiently determine whether polynomials of degree d satisfy $f(x_1, \ldots, x_n) \equiv g(x_1, \ldots, x_n)$

- Why not just expand the polynomials and see whether they are equal?

- How many terms can be in $(x_1 + x_2 + \cdots + x_n)^d$?
- Can be as large as $\binom{n}{d} \neq n^d$, which can be exponential in size

Polynomial Identity Testing

- It suffices to determine if $f(x_1, \ldots, x_n) - g(x_1, \ldots, x_n) \equiv 0$
- Determine whether a polynomial $P(x_1, \ldots, x_n) \equiv 0$
- Checking if a polynomial is identically zero has a large number of applications!

Graph Analysis

- Graphs can be represented via adjacency matrices
- The determinants of adjacency matrices (and other matrices) reveal information about the structural of the graph, e.g., whether the determinant is non-zero if and only if a bipartite graph has a perfect matching
- Determinants are polynomials!

Primality Checking

- The polynomial $P(z) \coloneqq (1+z)^n - 1 - z^n \pmod{n}$ is identically zero if and only if n is prime

Polynomial Identity Testing

- Determine whether a polynomial $P(x_1, \dots, x_n) \equiv 0$
- Expanding the polynomial can be slow, but evaluating the polynomial at any value of $x_1, \dots, x_n$ is efficient
- What should we do?

Polynomial Identity Testing

- Algorithm: Randomly pick values $x_1 = r_1, \ldots, x_n = r_n$ and evaluate $P(r_1, \ldots, r_n)$.
  - If $P(r_1, \ldots, r_n) = 0$, return $P(x_1, \ldots, x_n) = 0$
  - If $P(r_1, \ldots, r_n) \neq 0$, return $P(x_1, \ldots, x_n) \neq 0$

Polynomial Identity Testing

- Algorithm: Randomly pick values $x_1 = r_1, \ldots, x_n = r_n$ and evaluate $P(r_1, \ldots, r_n)$.
  - If $P(r_1, \ldots, r_n) = 0$, return $P(x_1, \ldots, x_n) = 0$
  - If $P(r_1, \ldots, r_n) \neq 0$, return $P(x_1, \ldots, x_n) \neq 0$

- If $P(x_1, \ldots, x_n) = 0$, then the protocol succeeds
- If $P(x_1, \ldots, x_n) \neq 0$, what is the probability of $P(r_1, \ldots, r_n) \neq 0$?

Polynomial Identity Testing

- If $P(x_1, \dots, x_n) = 0$, then the protocol succeeds
- If $P(x_1, \dots, x_n) \neq 0$, what is the probability of $P(r_1, \dots, r_n) \neq 0$?

- Suppose we choose x_i randomly from $\{1, \dots, S\}$
- By Schwartz-Zippel, the probability that $P(r_1, \dots, r_n) \neq 0$ is at least $1 - \frac{d}{S} \geq 0.9$ for $S \geq 10d$

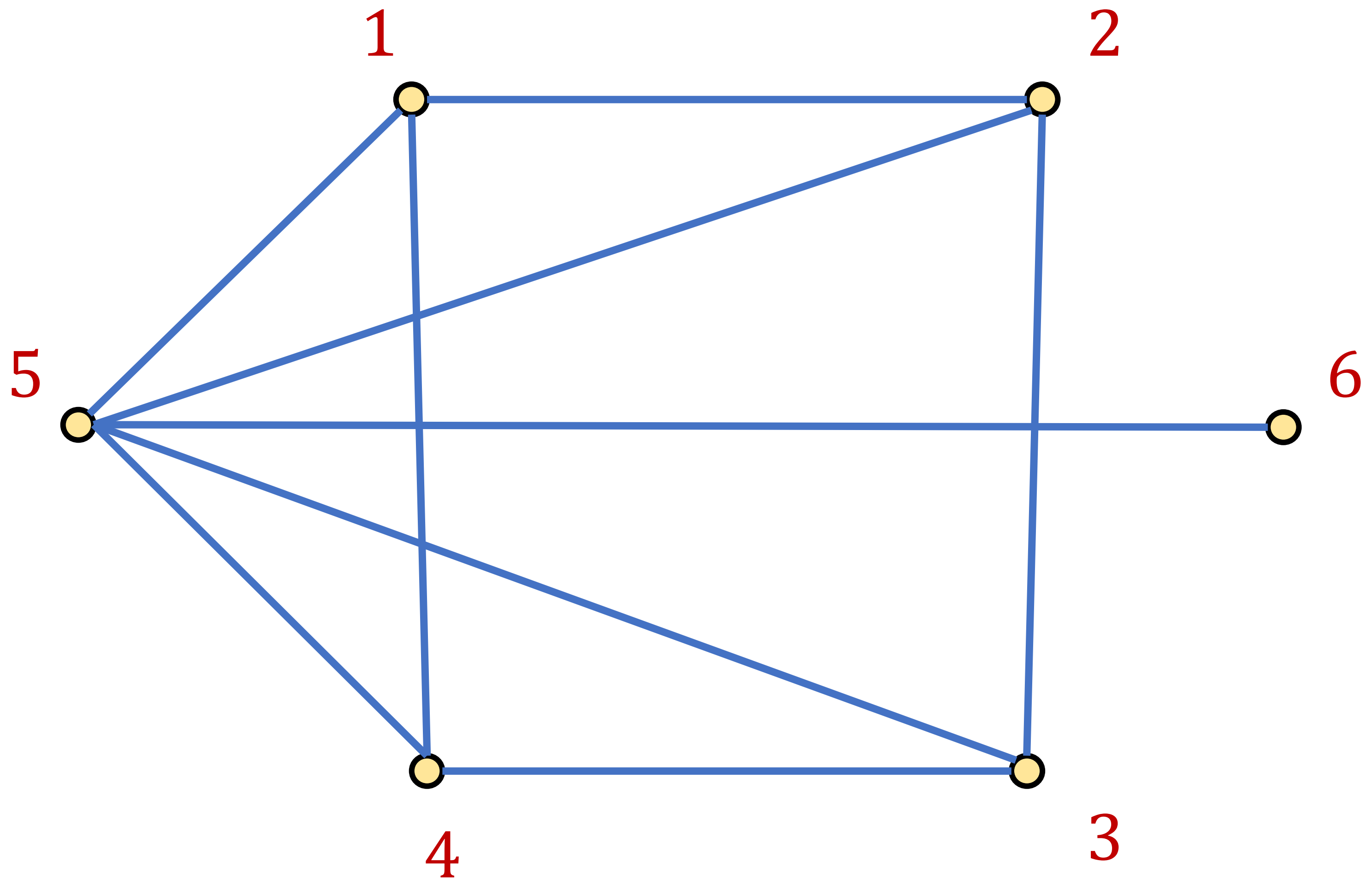
Questions?

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

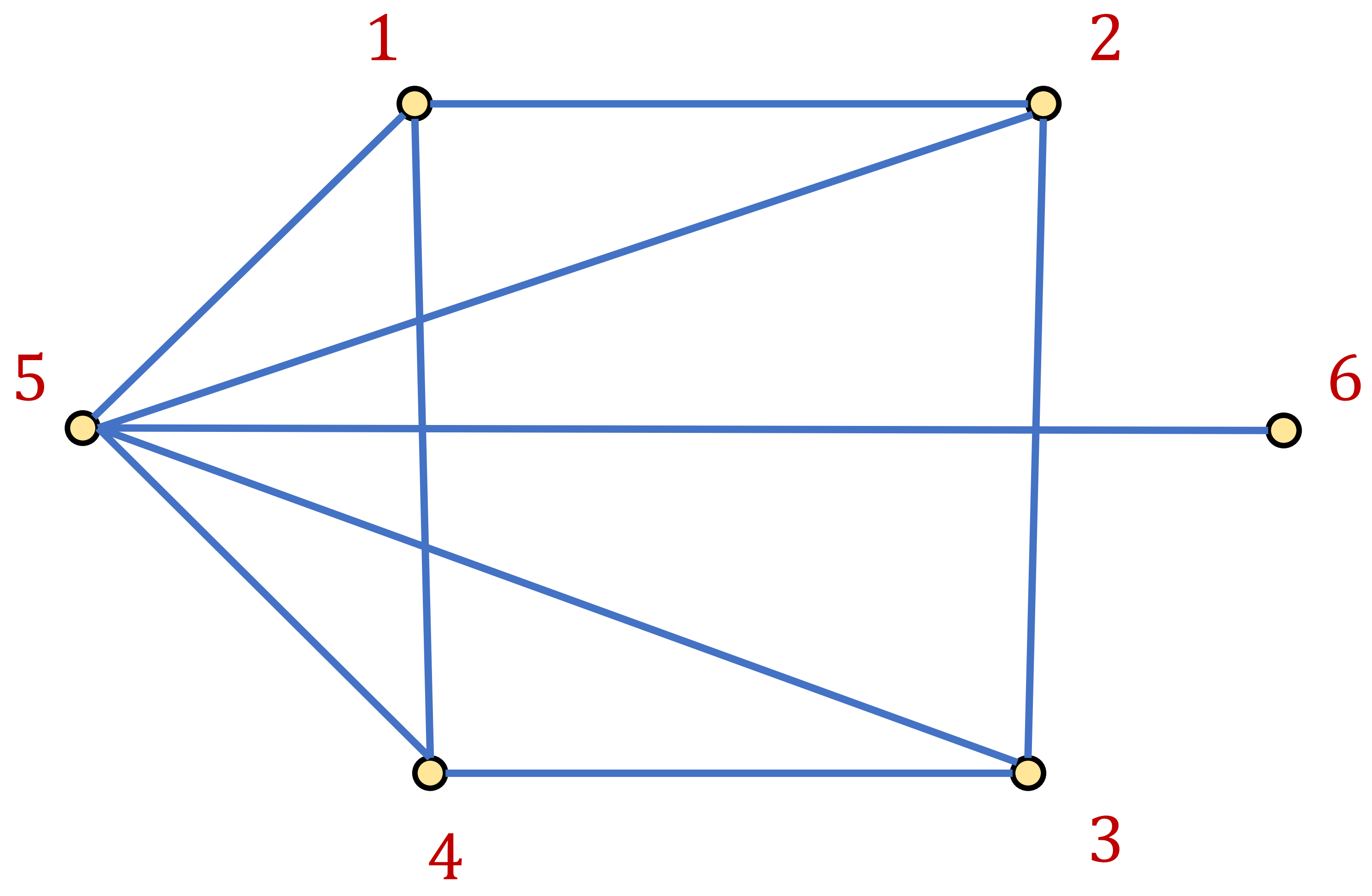

1
2
5
6
4
3

Cuts

- A cut $C = S_1, S_2$ of a graph G is a partition of the vertices V into a set S_1 and the remaining vertices $S_2 = V - S_1$

- An edge (u, v) crosses the cut C if $u \in S_1$ and $v \in S_2$

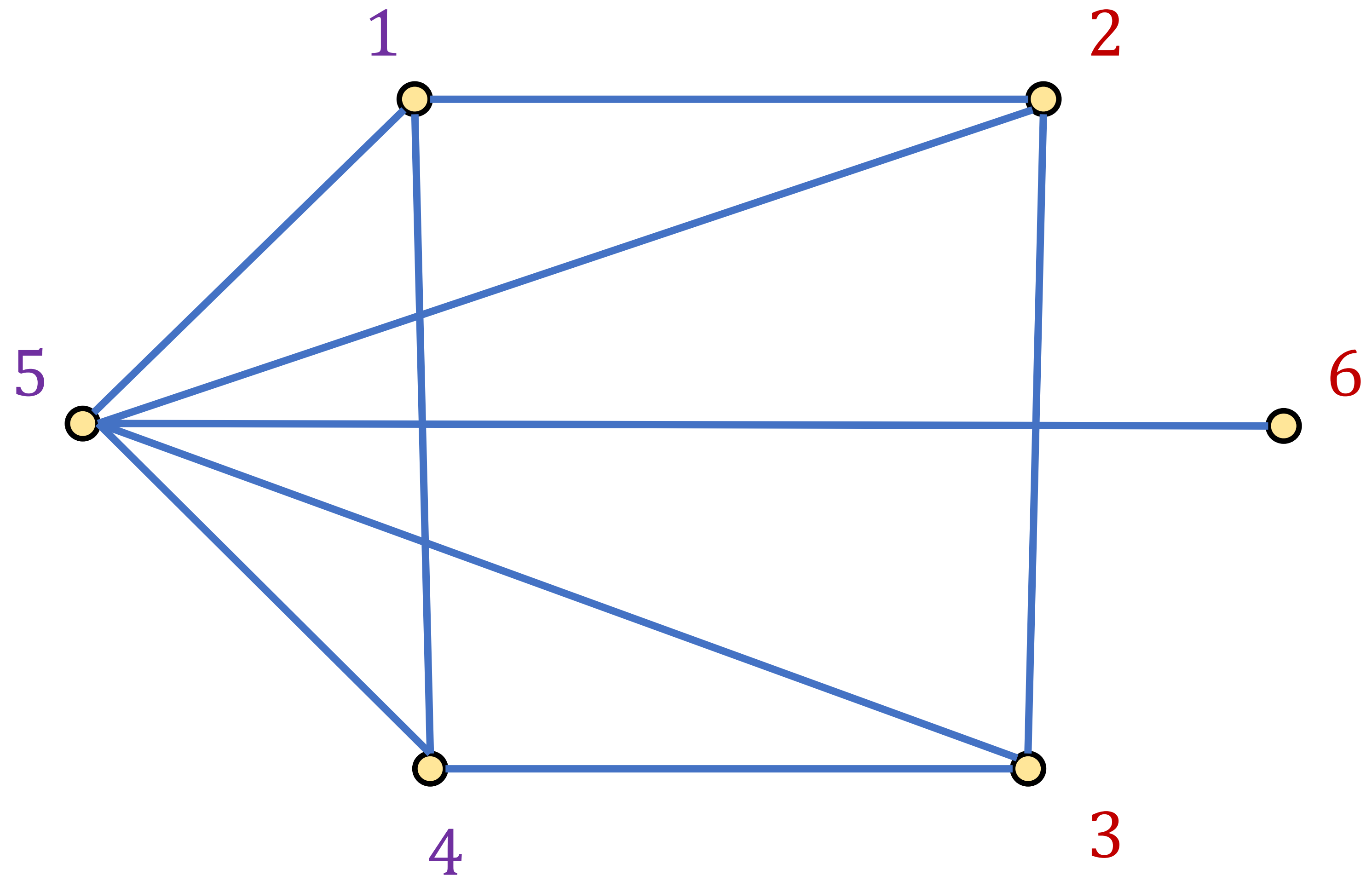
- The size of the cut C is the number of edges that cross C


1
2
5
6
4
3

What is the size of the cut $C = S_1, S_2$?

$S_1 = \{1,4,5\}$

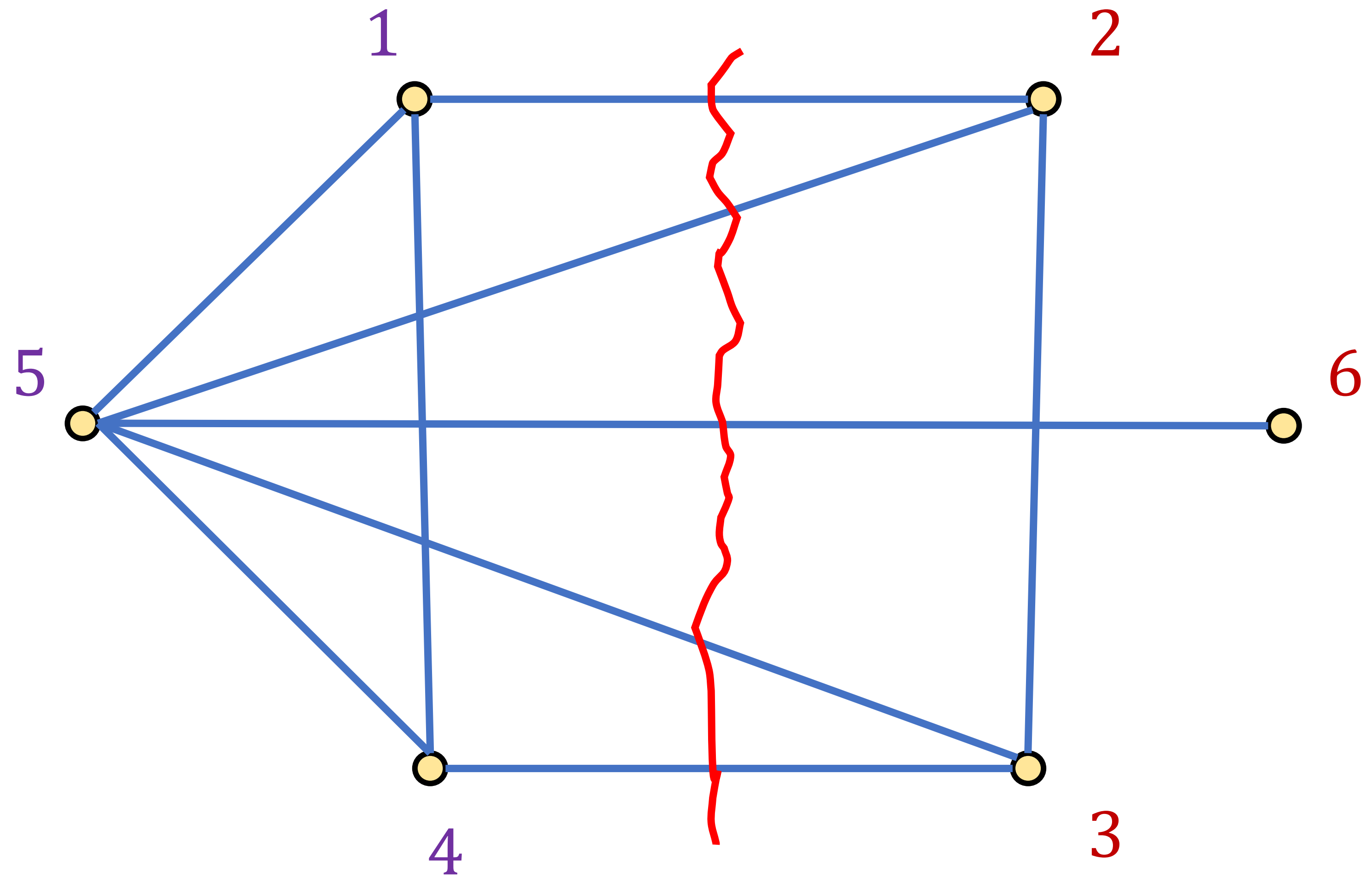
$S_2 = \{2,3,6\}$



What is the size of the cut $C = S_1, S_2$?

$S_1 = \{1,4,5\}$

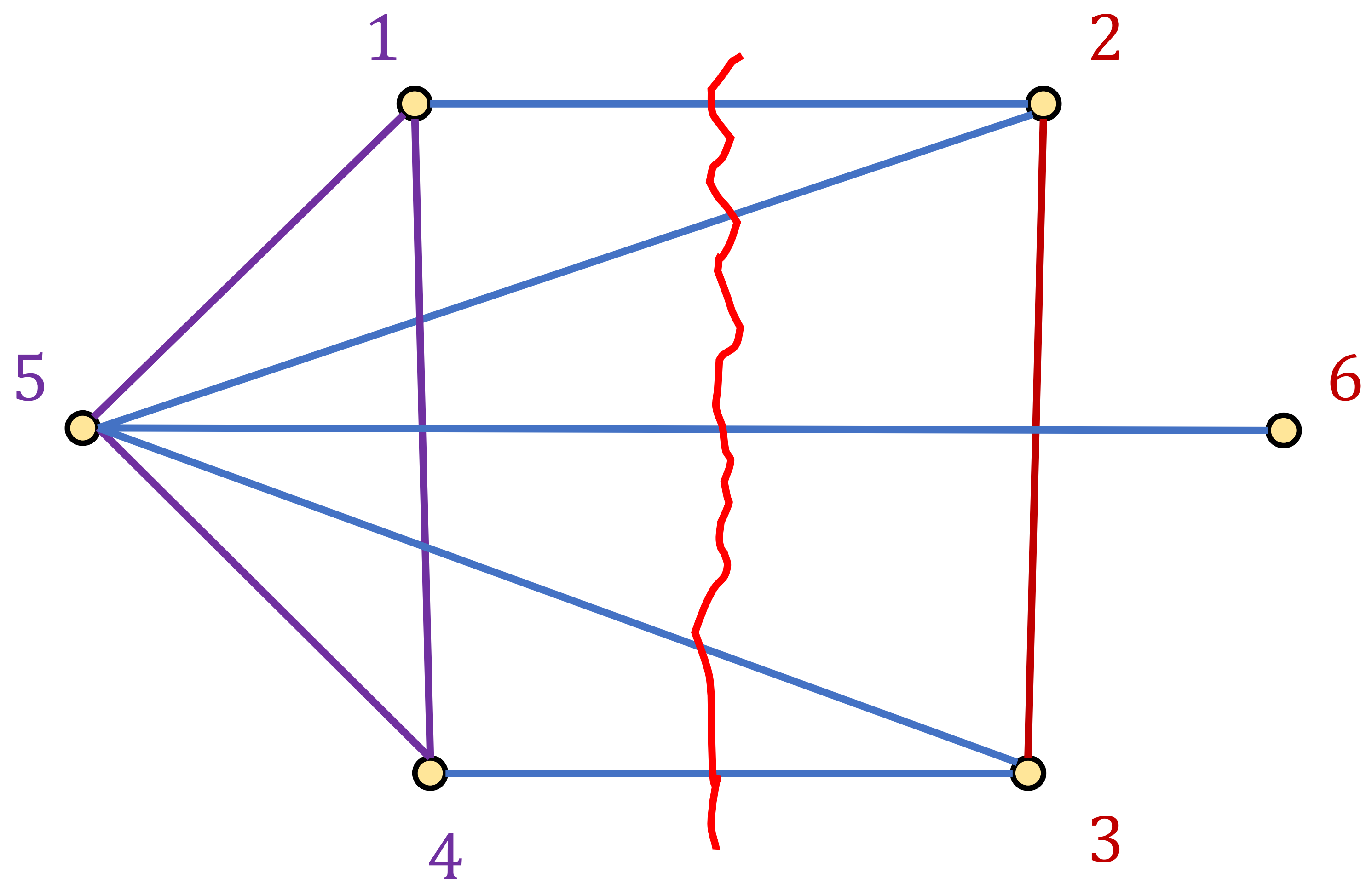
$S_2 = \{2,3,6\}$



What is the size of the cut $C = S_1, S_2$?

$S_1 = \{1,4,5\}$

$S_2 = \{2,3,6\}$



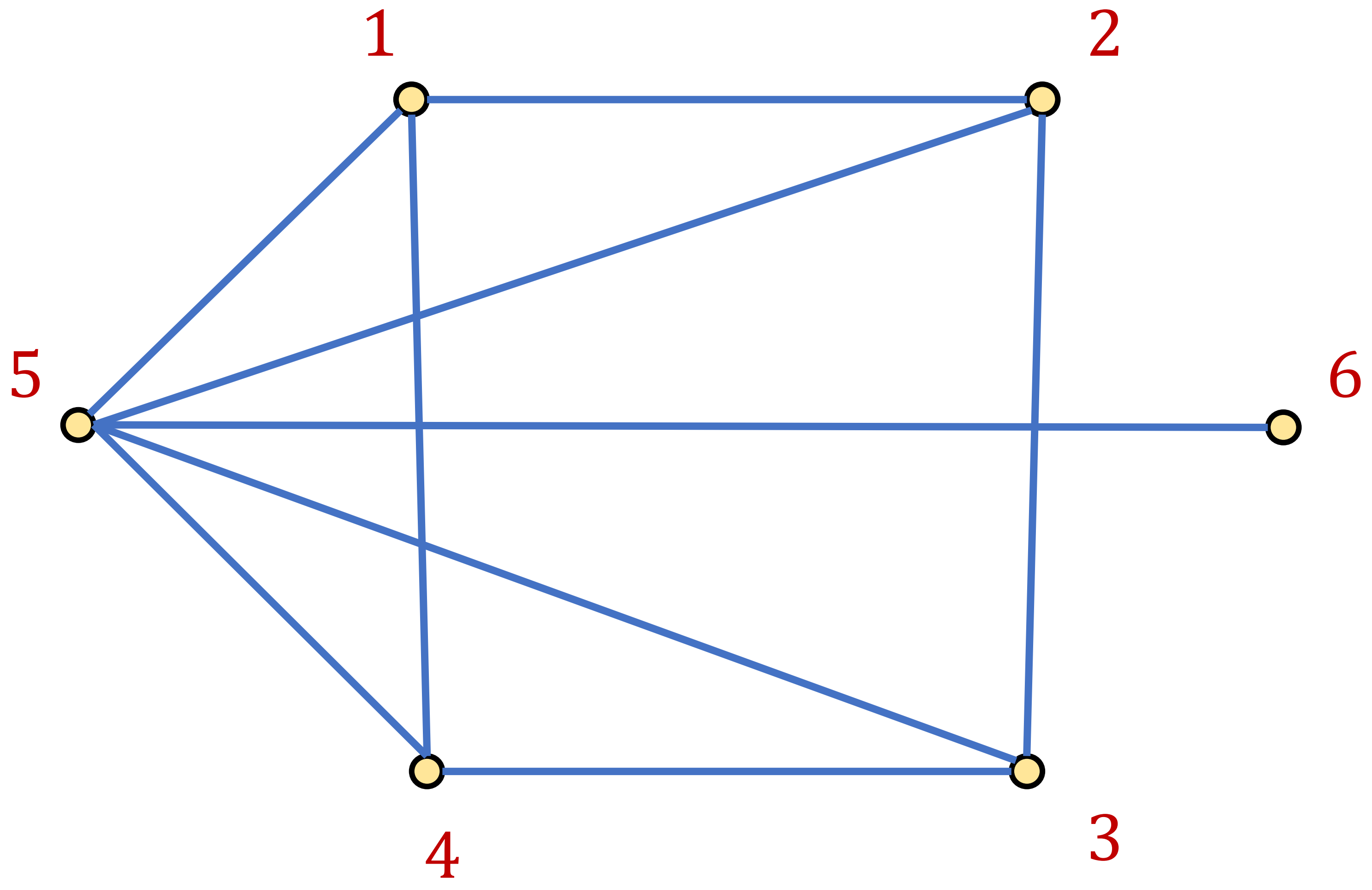
The cut size is five

Minimum Cut

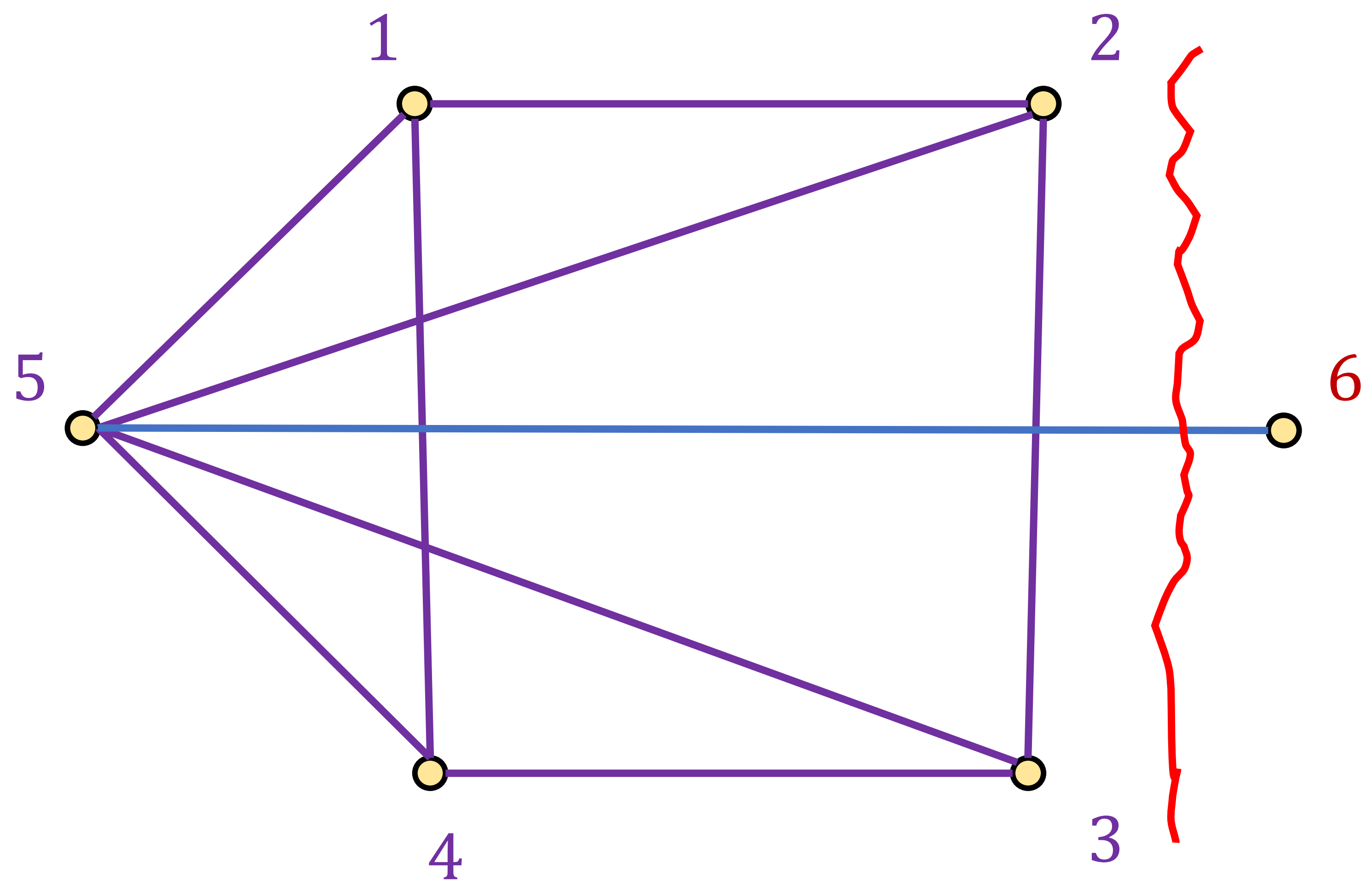
- The minimum cut of a graph is the size of the smallest cut across all pairs of sets of vertices S_1 and $S_2 = V - S_1$

- Find the minimum cut of a graph G

What is the minimum cut of the graph?

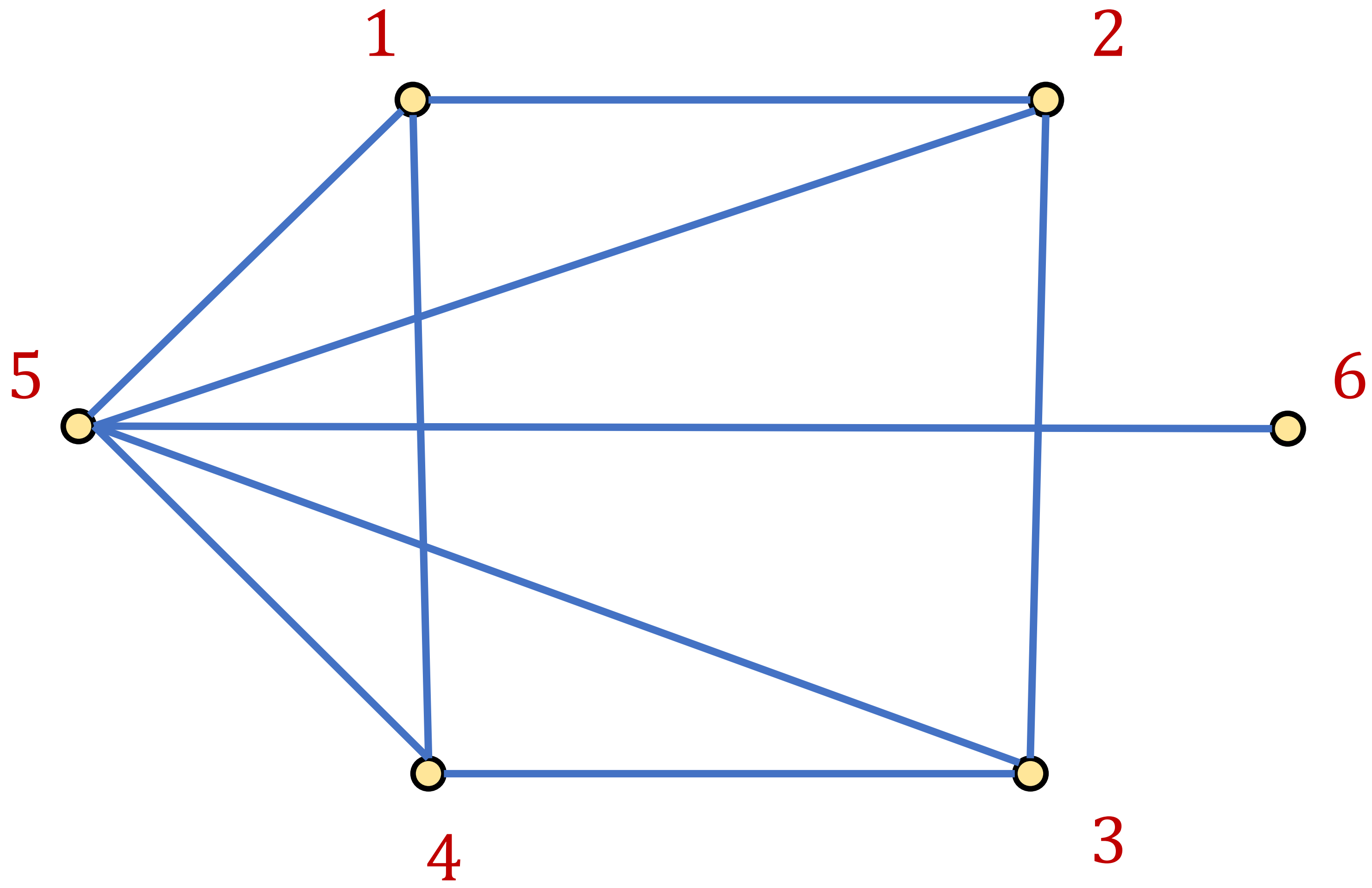


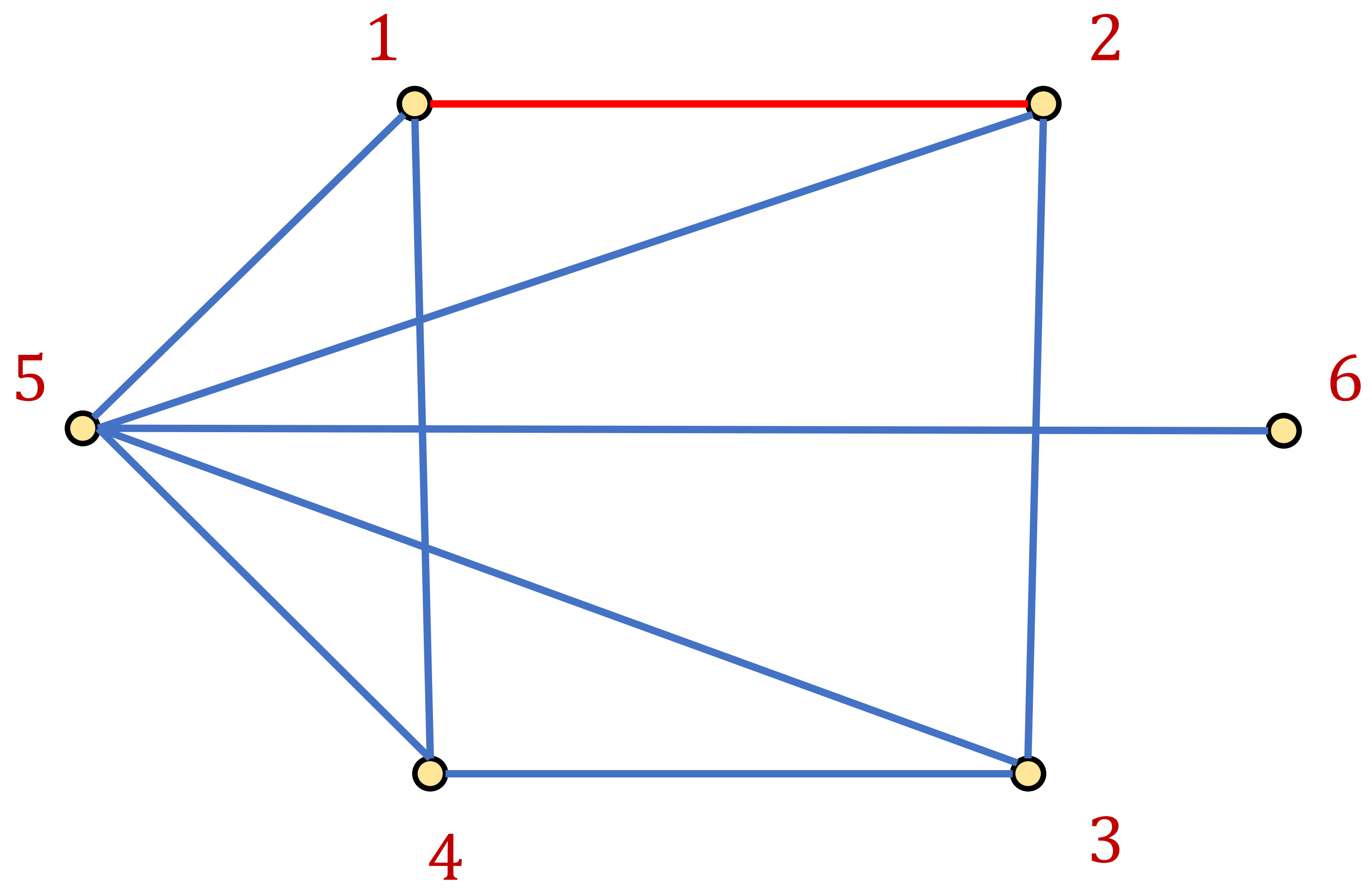
What is the minimum cut of the graph?

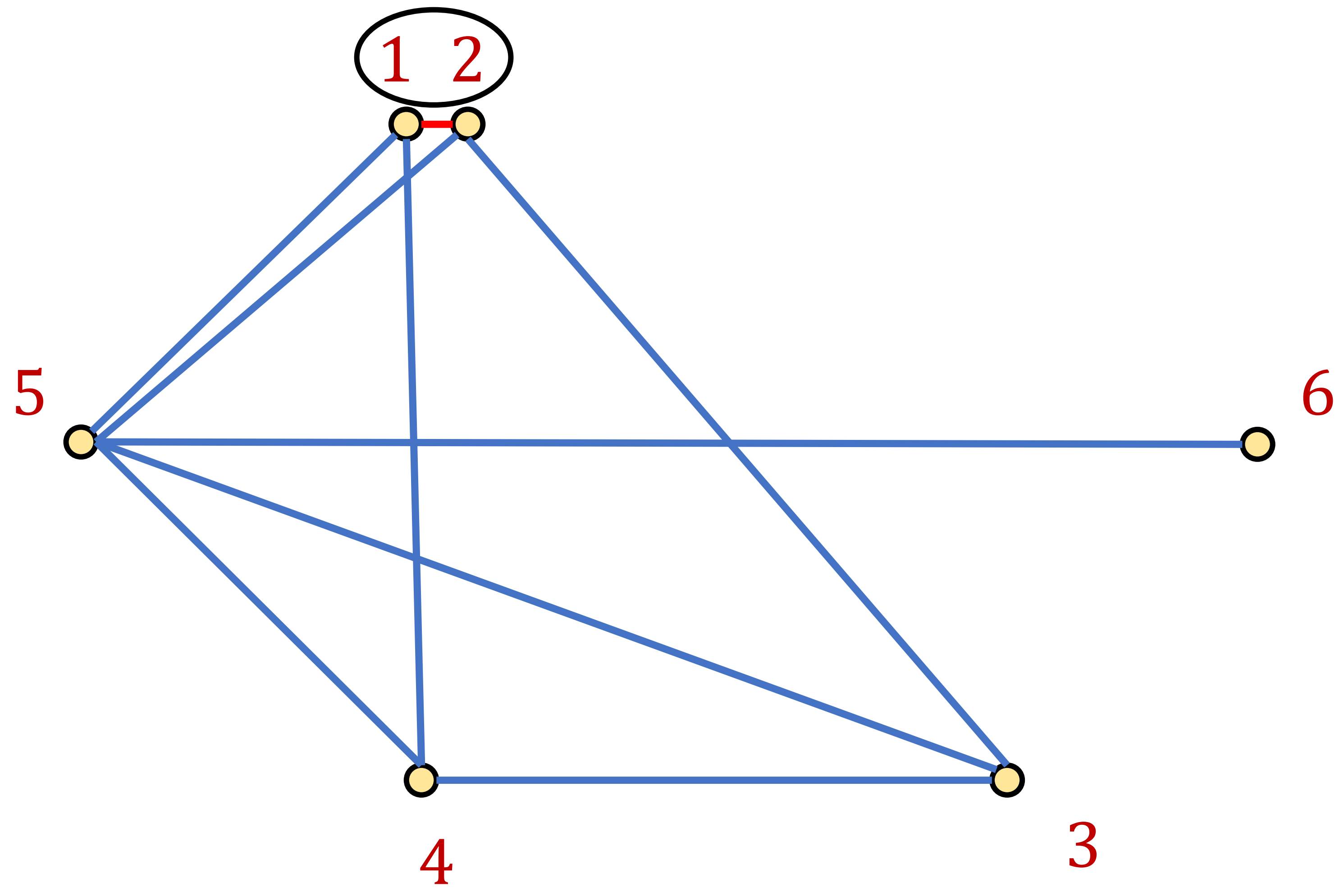


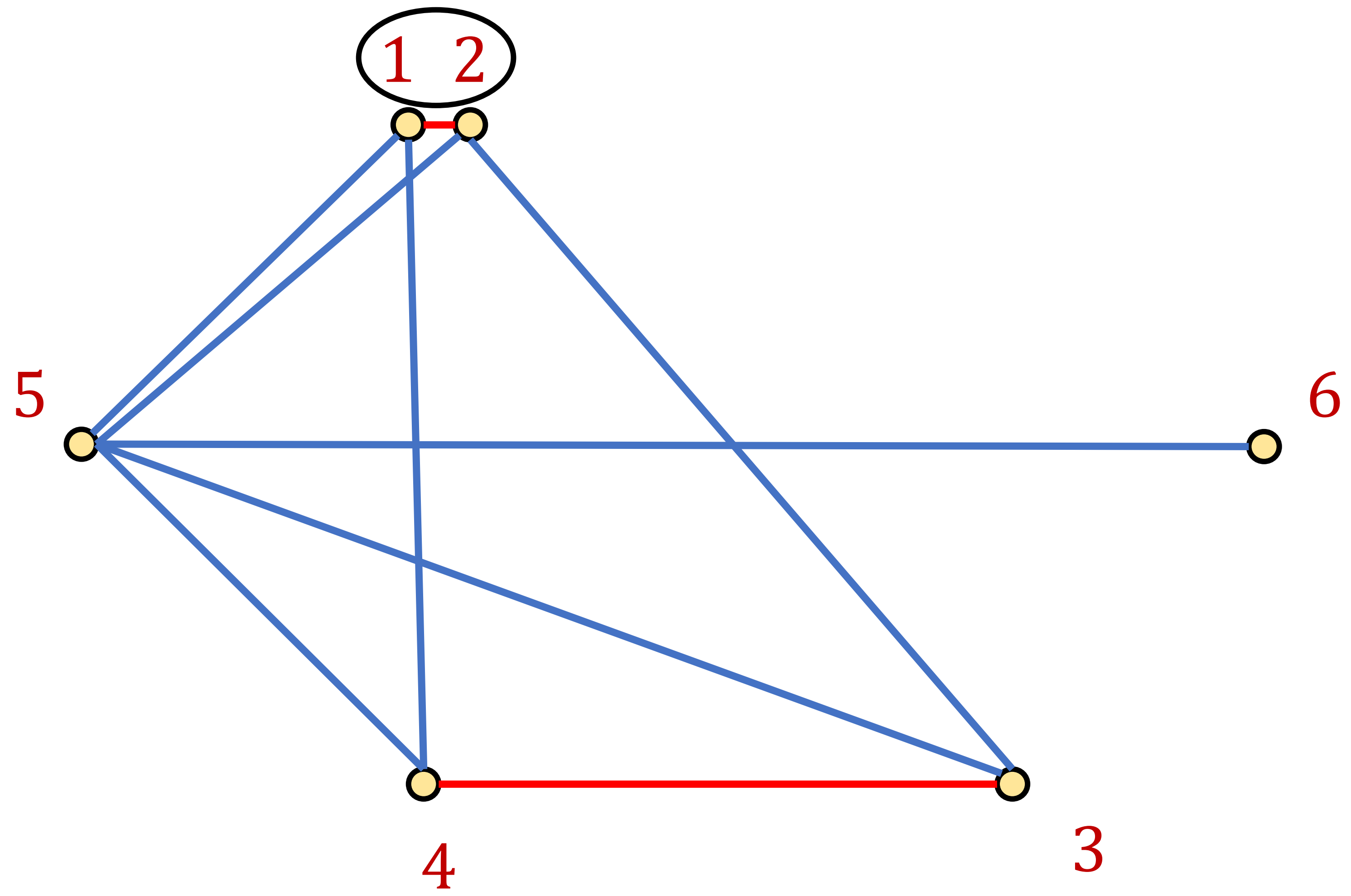
Karger's Minimum Cut Algorithm

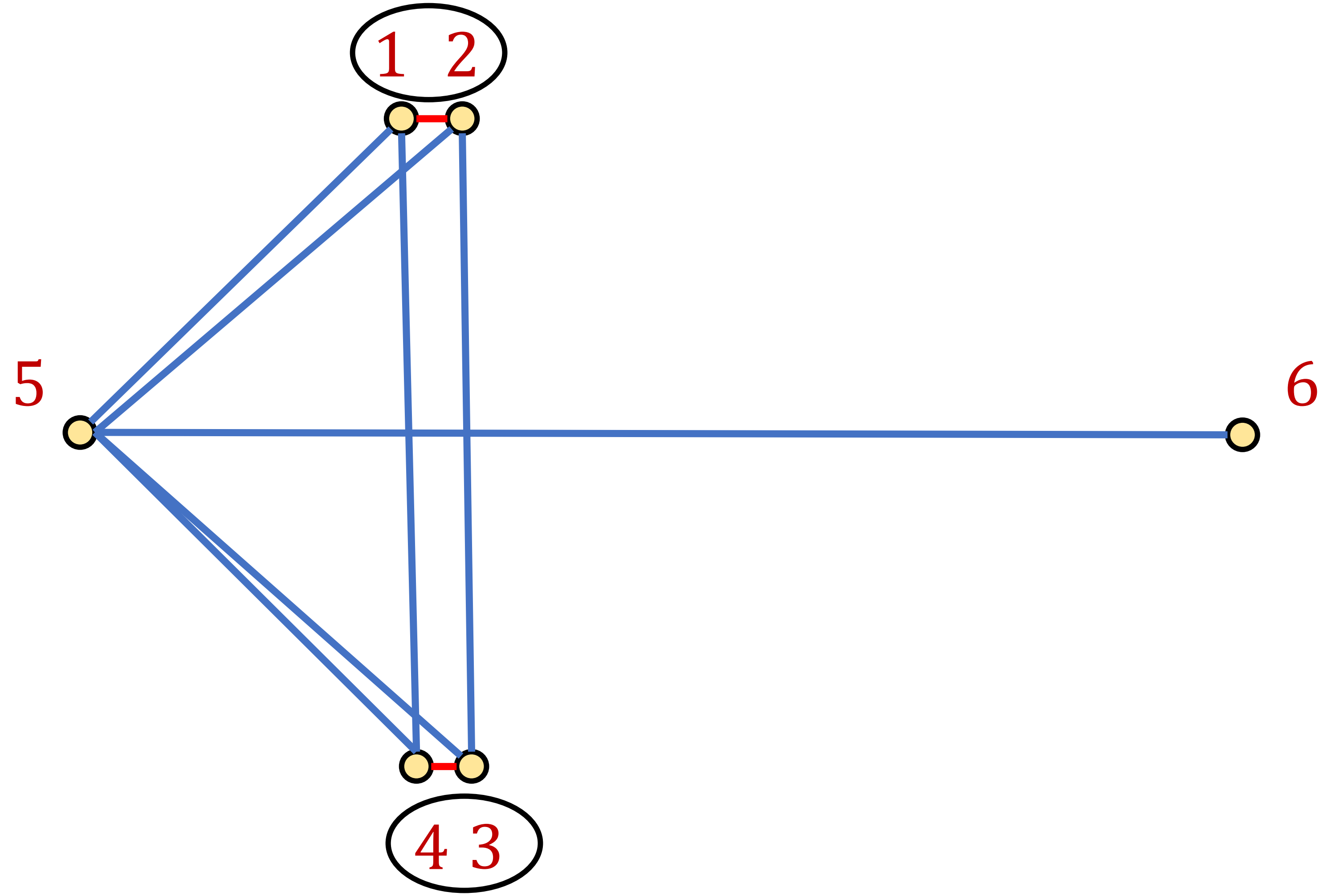
1. Start with original graph and iteratively reduce the number of vertices via a series of edge contractions
2. In each step, choose a random edge and merge the two endpoints of that edge into a single vertex, preserving edges (allow multi-edges but not self-loops)
3. Iterate until there are only two vertices u_1 and u_2 left
4. Return the vertices merged into u_1 as one set
5. Return the vertices merged into u_2 as the other set

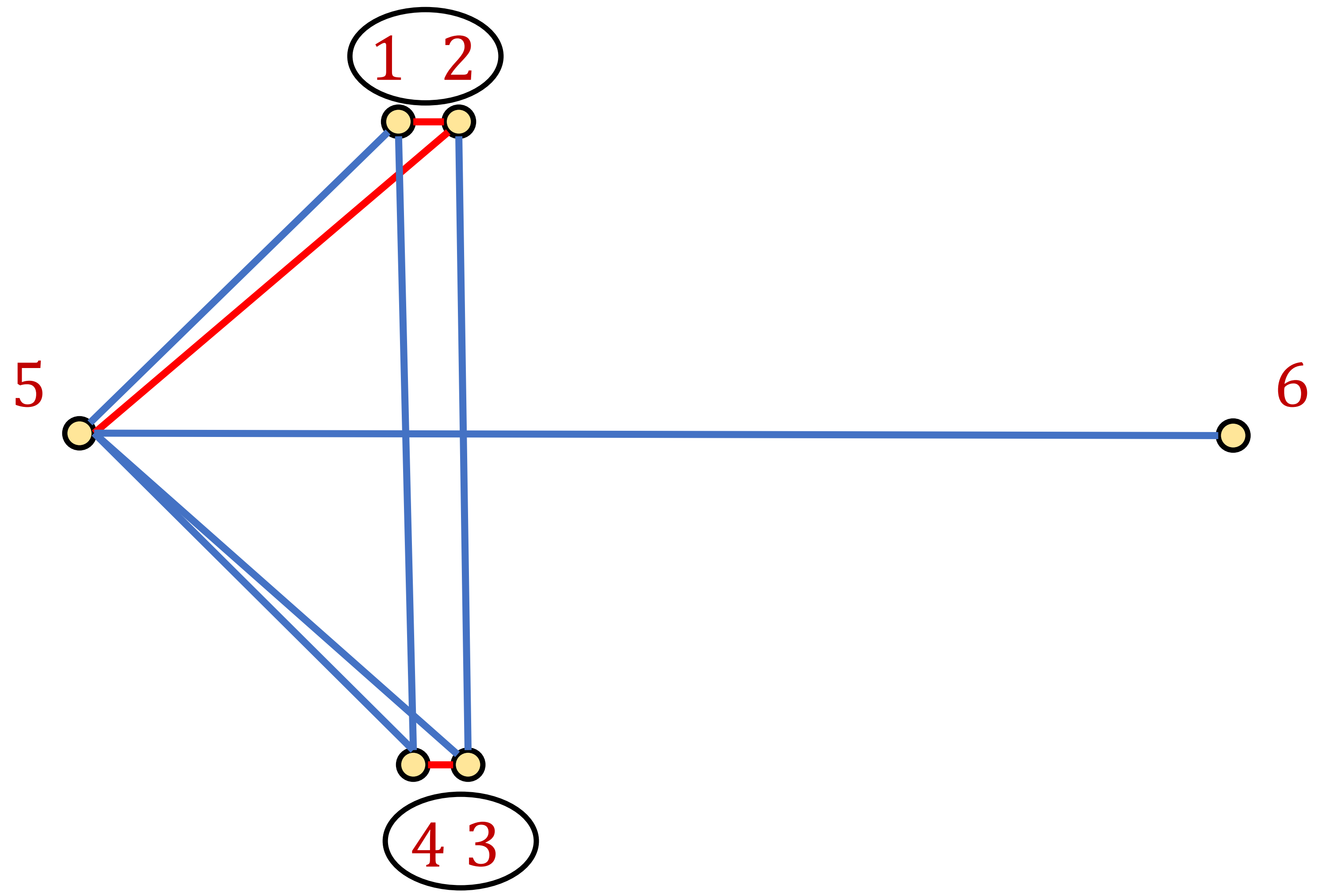

1
2
5
6
4
3

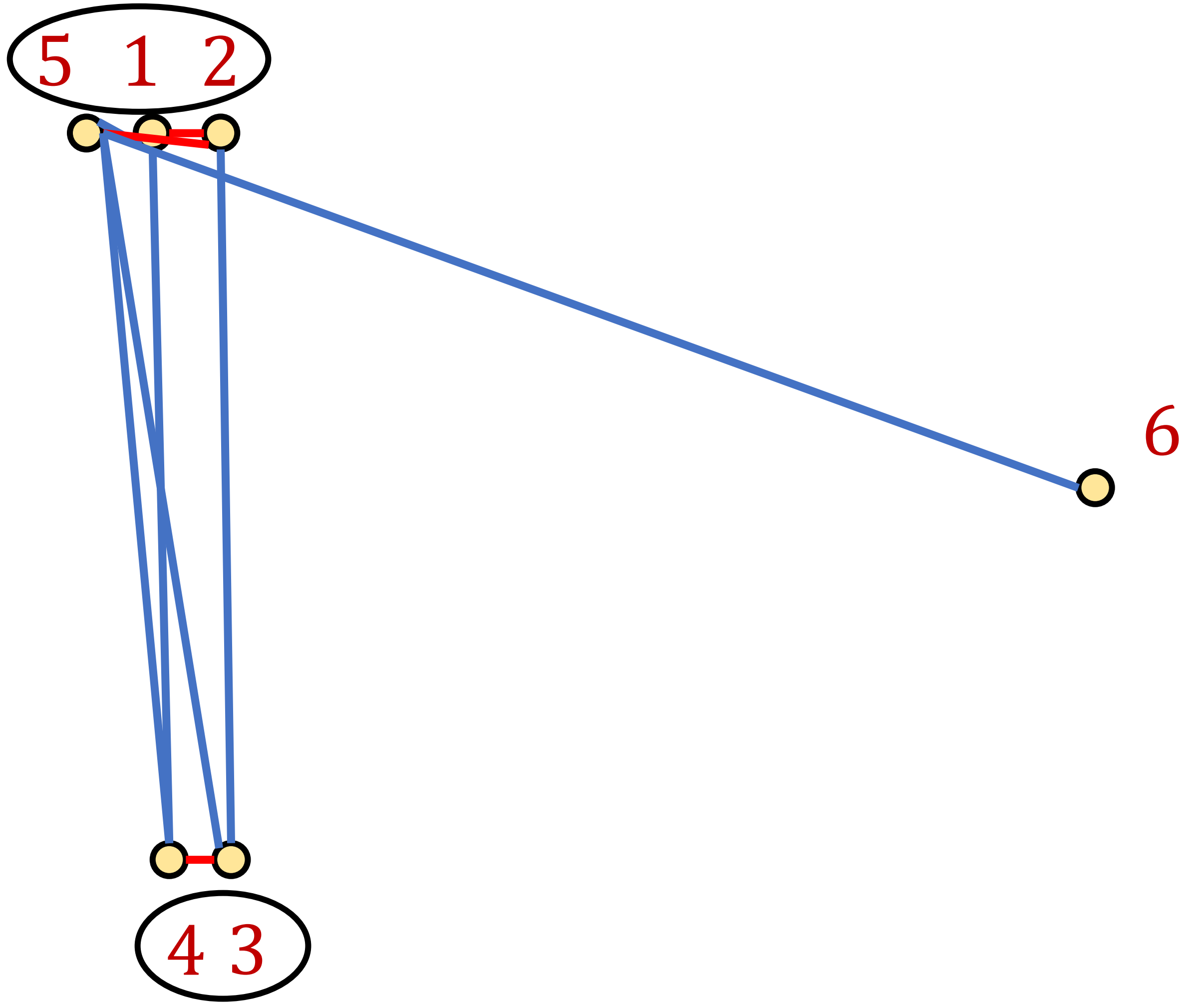

1
2
5
6
4
3

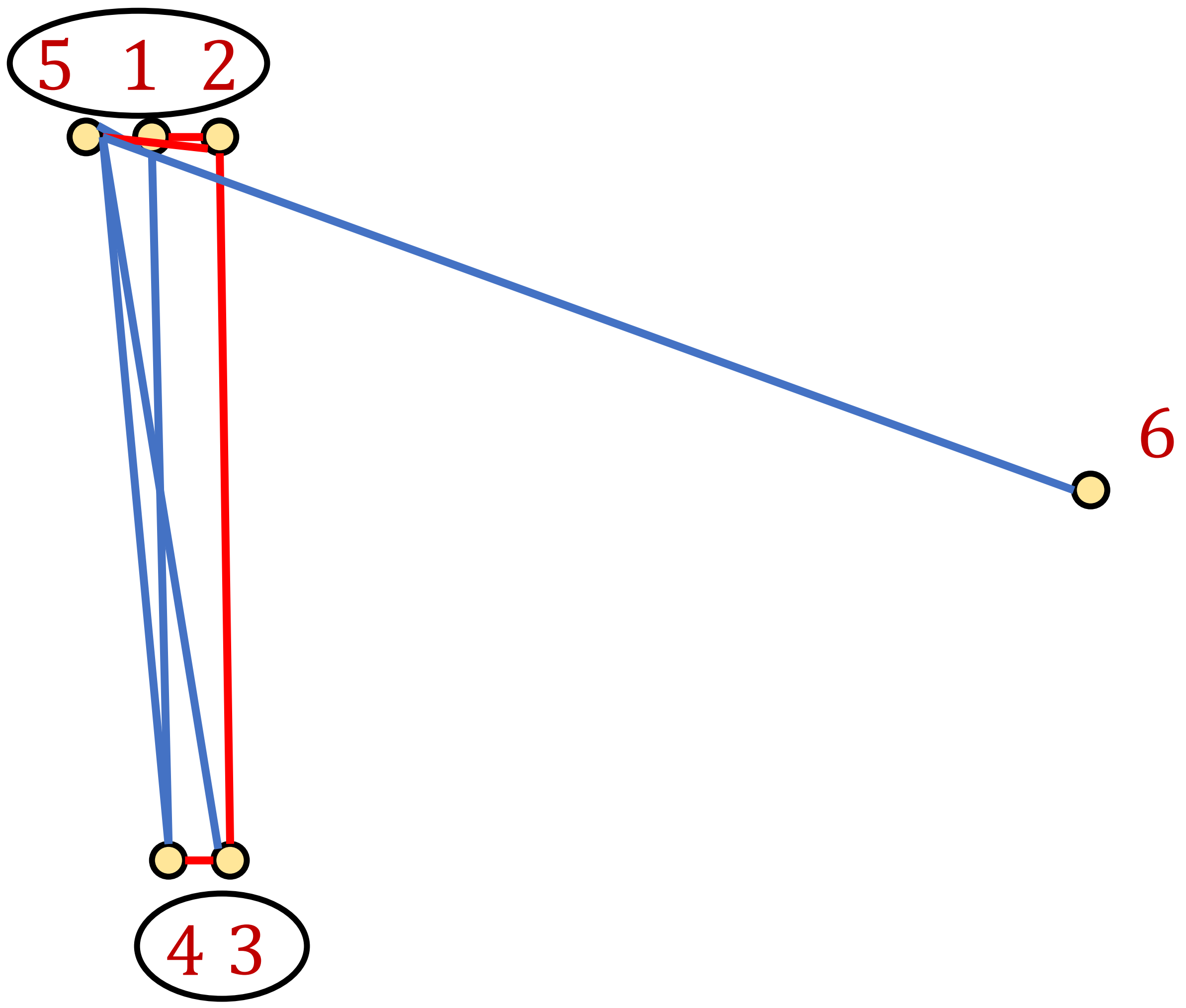

1
2
5
6
4
3

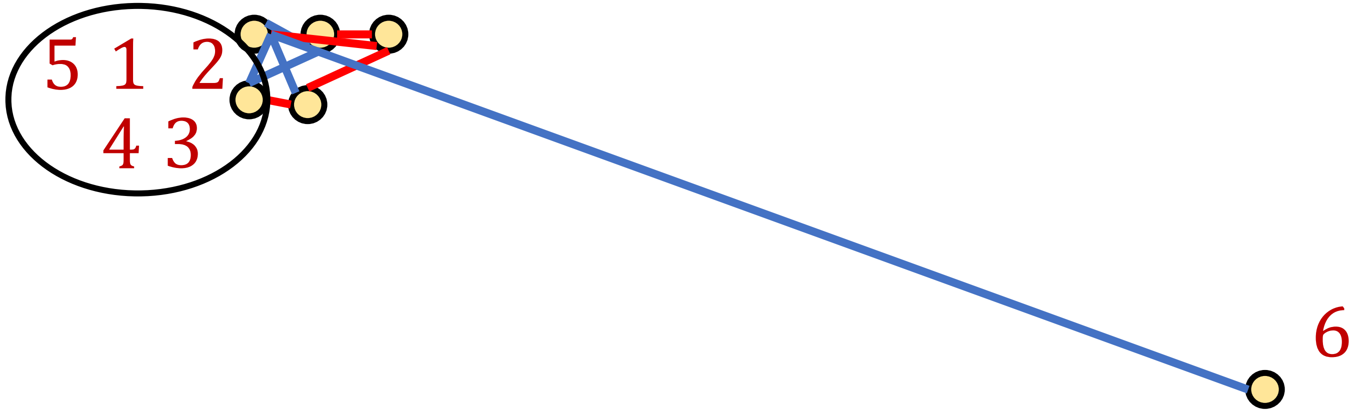

1 2
5
6
4
3


1 2
5
6
4 3


1 2
5
6
4 3

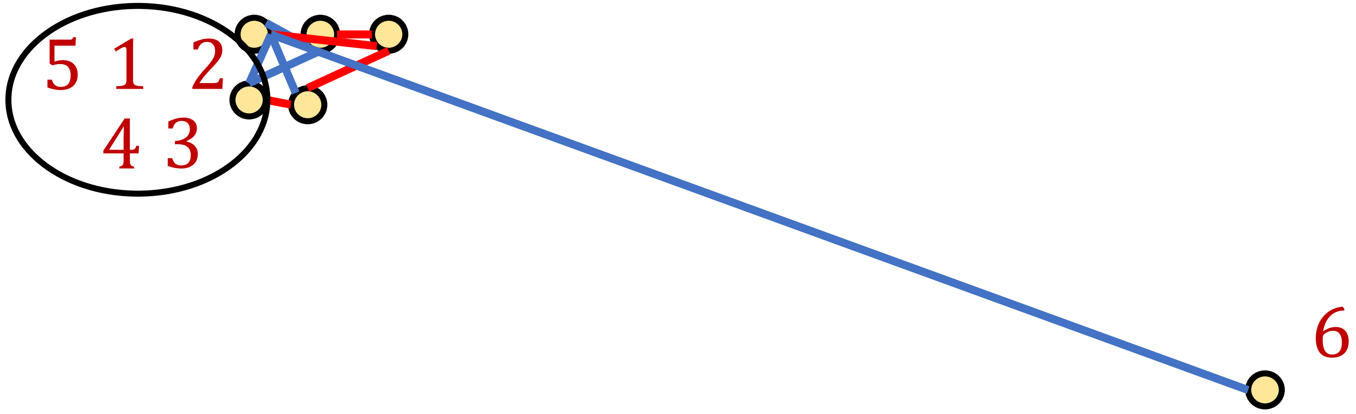

5 1 2
6
4 3


5 1 2
6
4 3


5 1 2
4 3
6

$S_1 = \{1,2,3,4,5\}$

$S_2 = \{6\}$



Return S_1, S_2

Karger’s Minimum Cut Algorithm

- Intuition: Suppose the graph is disconnected. Then we will ALWAYS return the correct min-cut
- Now suppose the graph consists of two components connected by a single. Algorithm is successful as long as it avoids selecting the single edge that crosses the two components
- Why? As long as it avoids the single edge, each edge contraction will just shrink one of the two components
- There is a good chance we never contract the single edge

Karger’s Minimum Cut Algorithm

- Analysis: Fix a min-cut $C = S_1, S_2$ with size k
- Probability that we contract an edge of C is $\frac{k}{|E|}$, where $|E|$ is the number of edges
- Since the min-cut is k, then each vertex must have degree at least k so $|E| \geq \frac{nk}{2}$
- The probability that we DO NOT contract an edge of C is at least $1 - \frac{k}{(nk/2)} = \frac{n-2}{n}$

Karger's Minimum Cut Algorithm

- After i steps, the number of vertices left is $n - i$, so the probability that we DO NOT contract an edge of C is at least $\frac{n-i-2}{n-i}$

- Probability of success is at least:

$$\frac{n-2}{n} \times \frac{n-3}{n-1} \times \frac{n-4}{n-2} \times \cdots \times \frac{1}{3} \geq \frac{2}{n(n-1)}$$

Karger’s Minimum Cut Algorithm

- Probability of success is at least $\frac{2}{n^2}$
- Will succeed with probability 0.99 if we repeat $O(n^2)$ times