

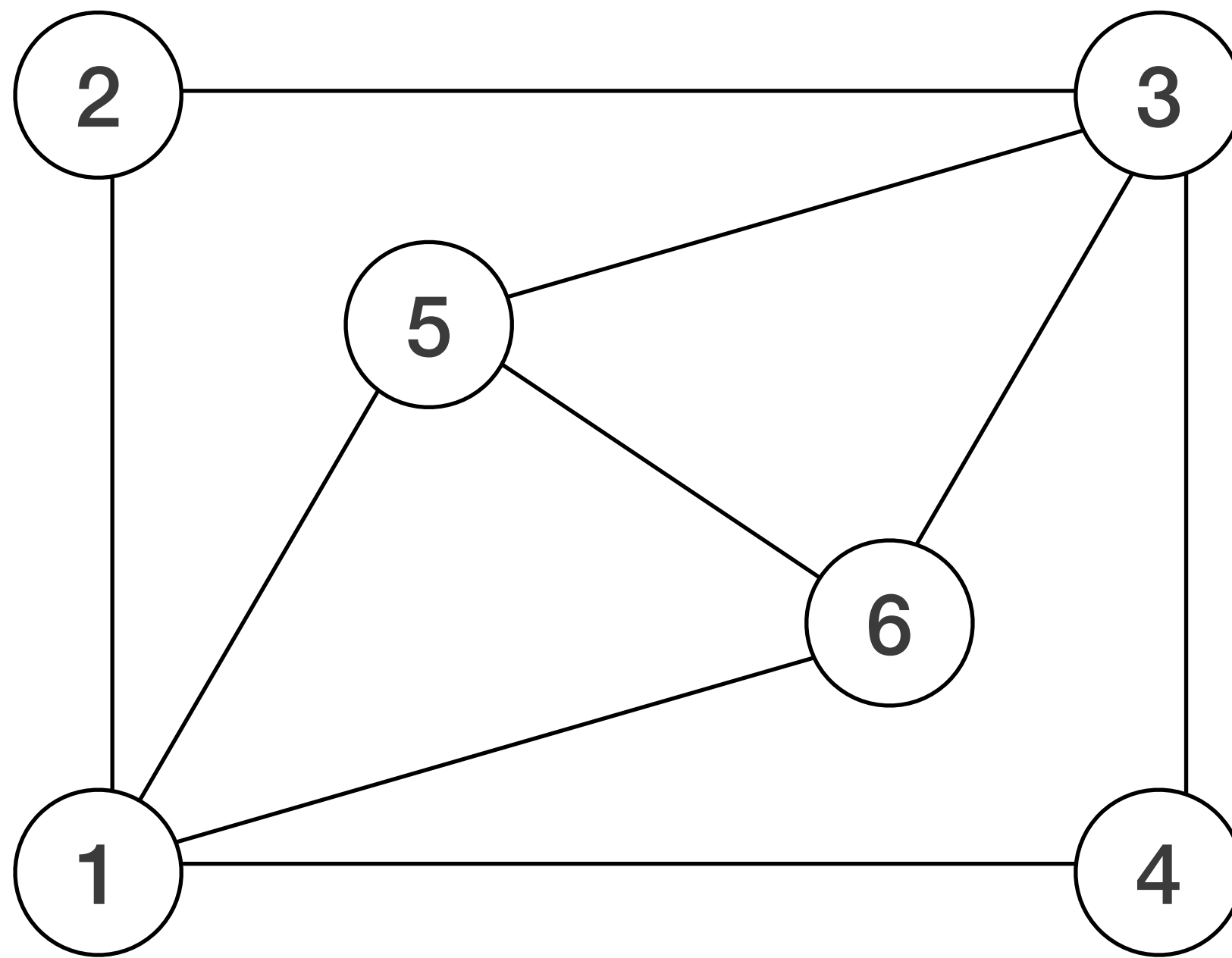
Dynamic Graph Structures

Keerti Choudhary

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:

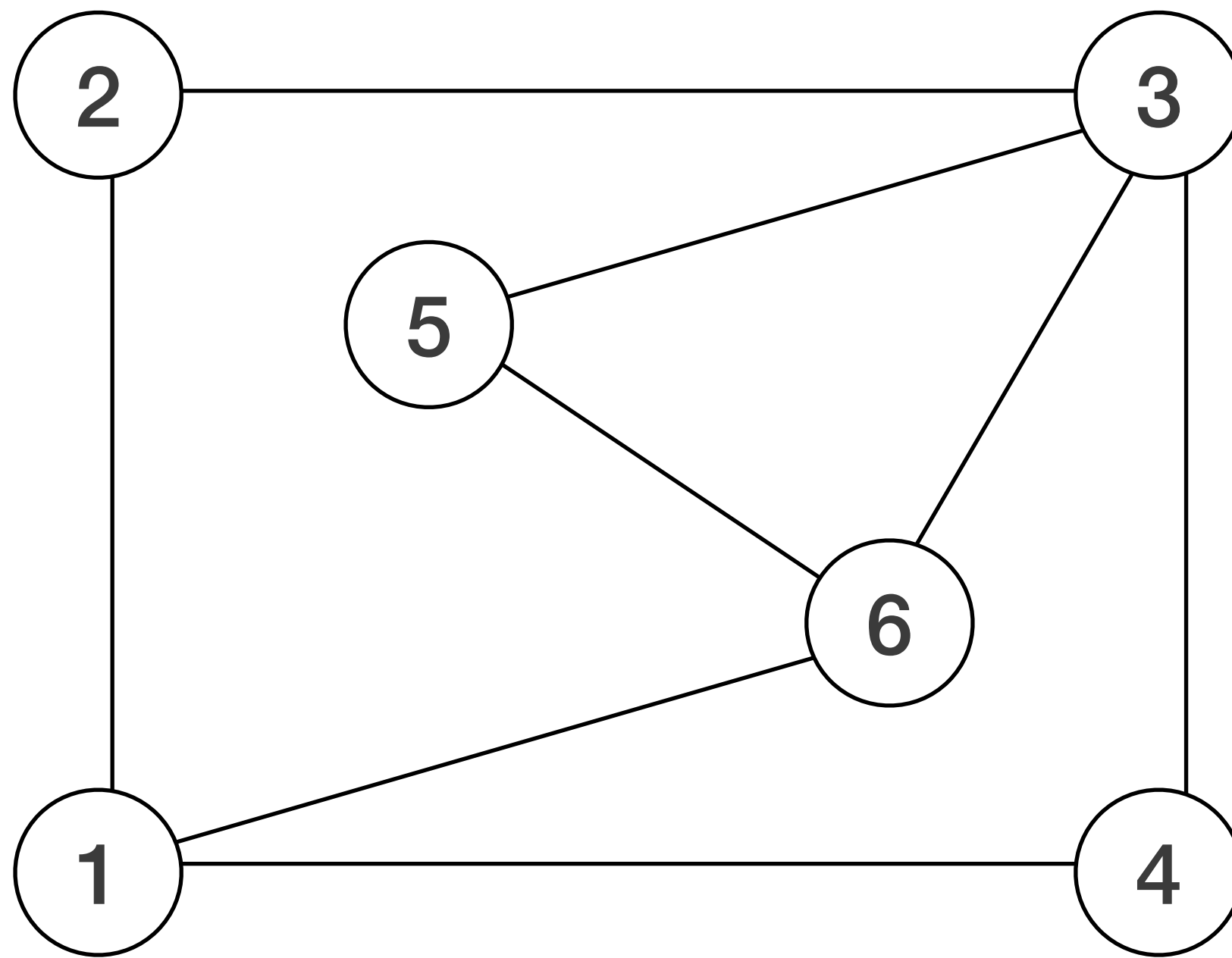


Time: $t = 0$

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:

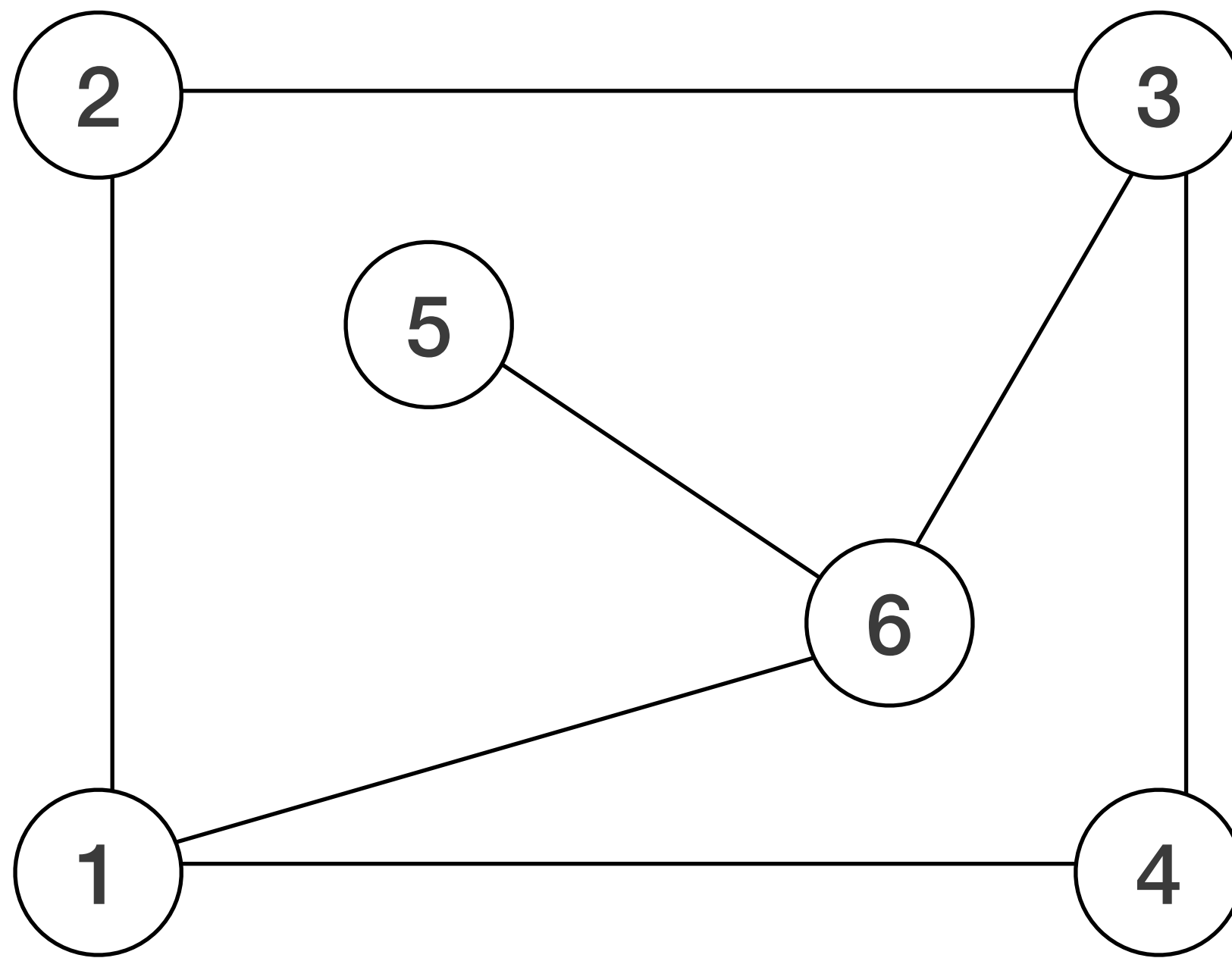


Time: $t = 1$

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:

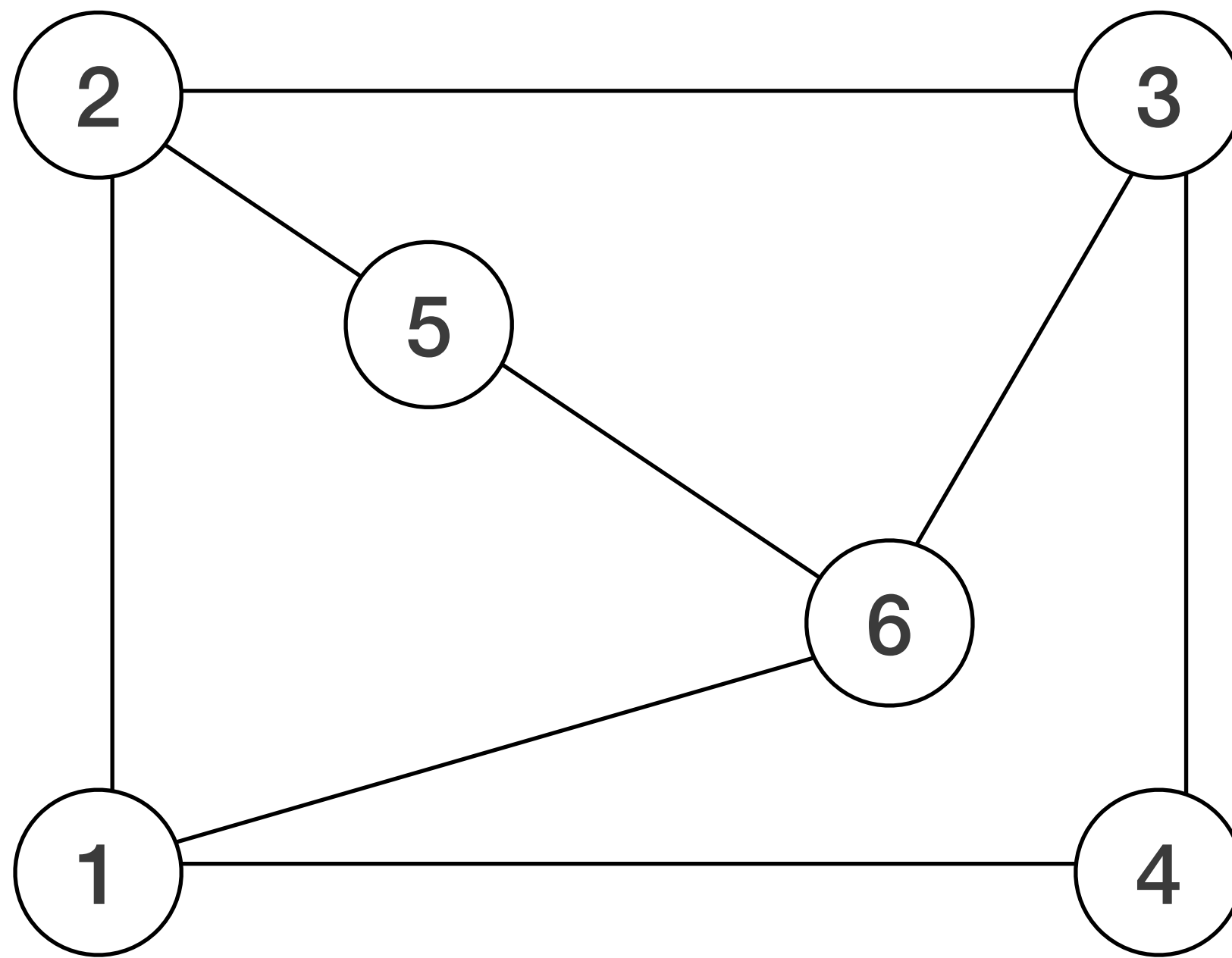


Time: $t = 2$

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:

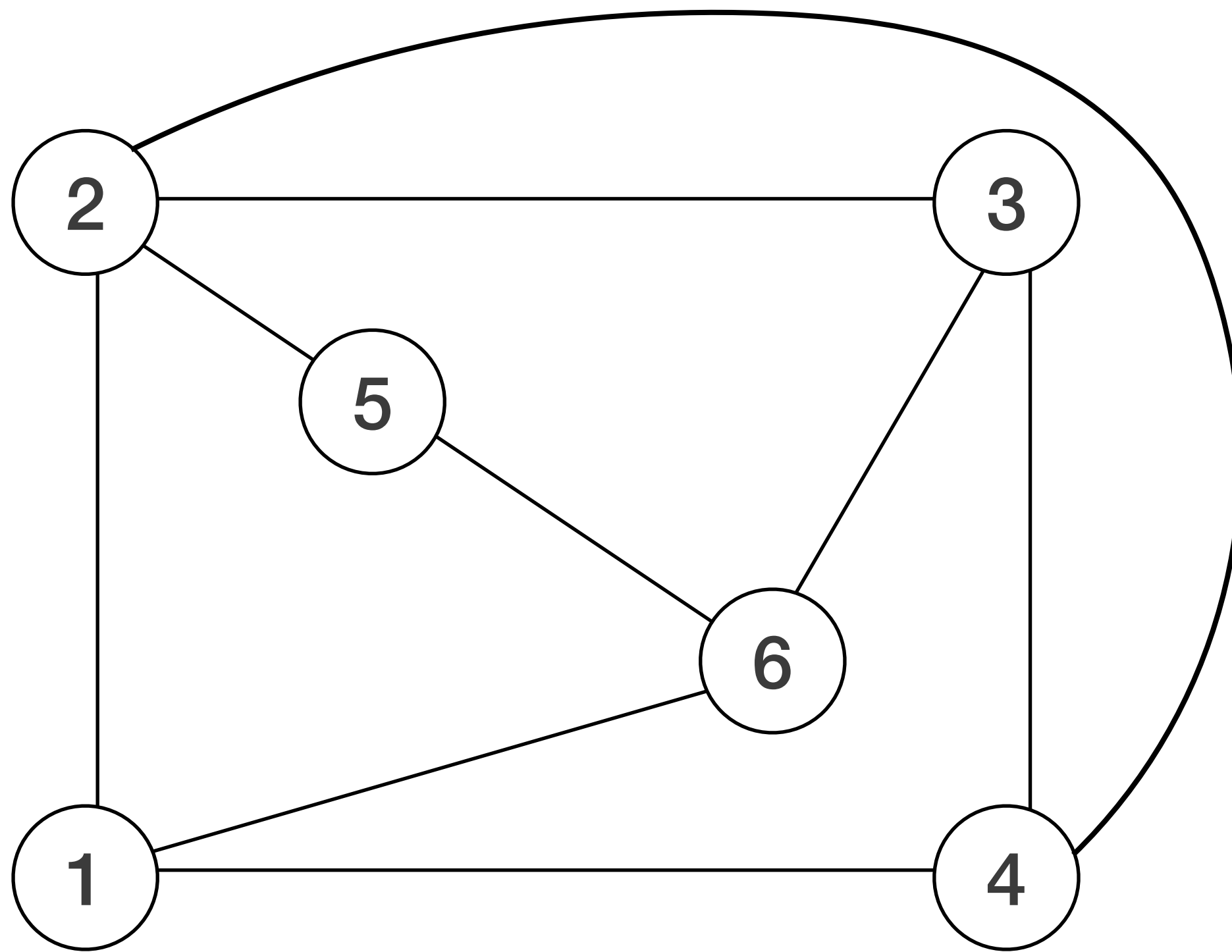


Time: $t = 3$

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:

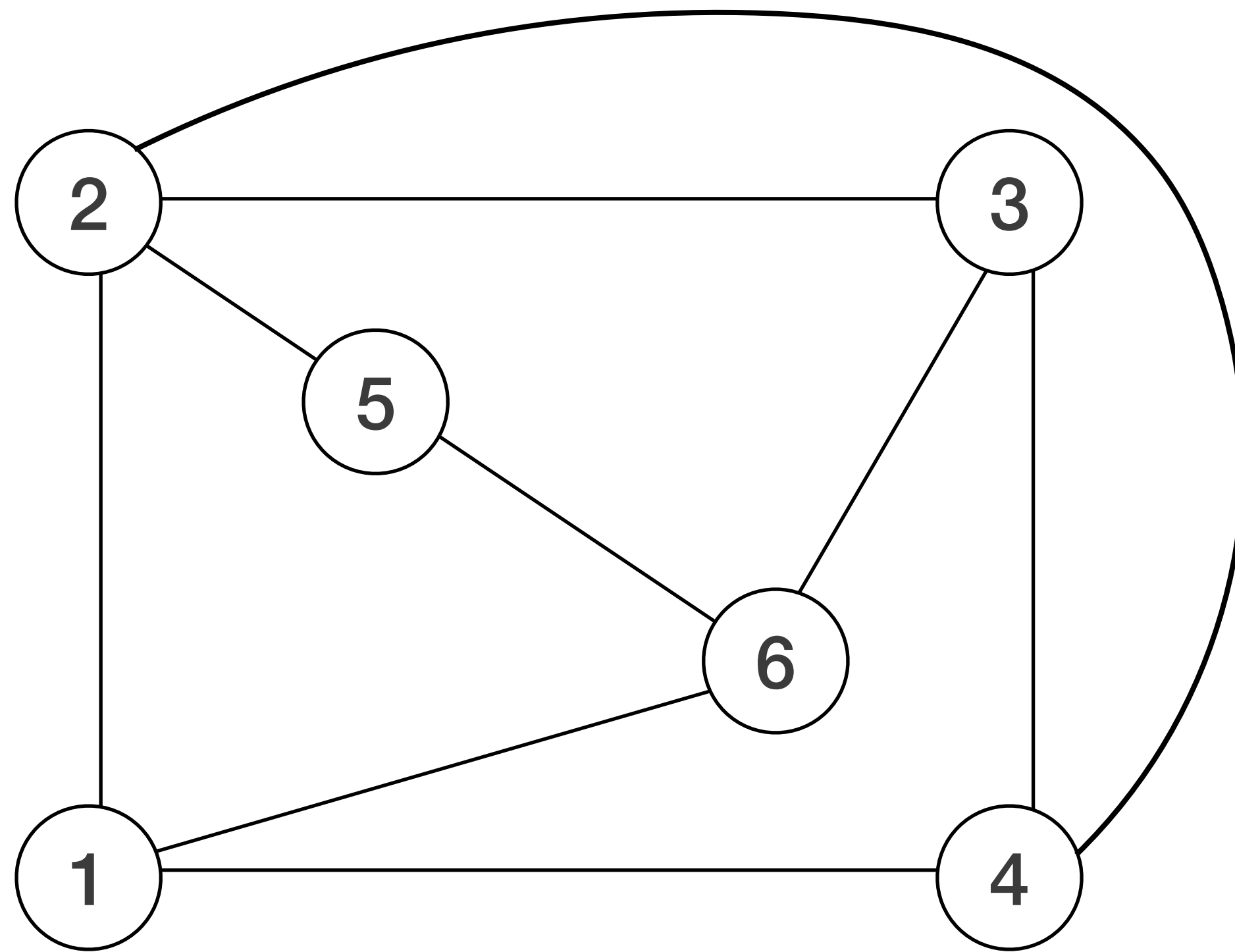


Time: $t = 4$

What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:



Time: $t = 4$

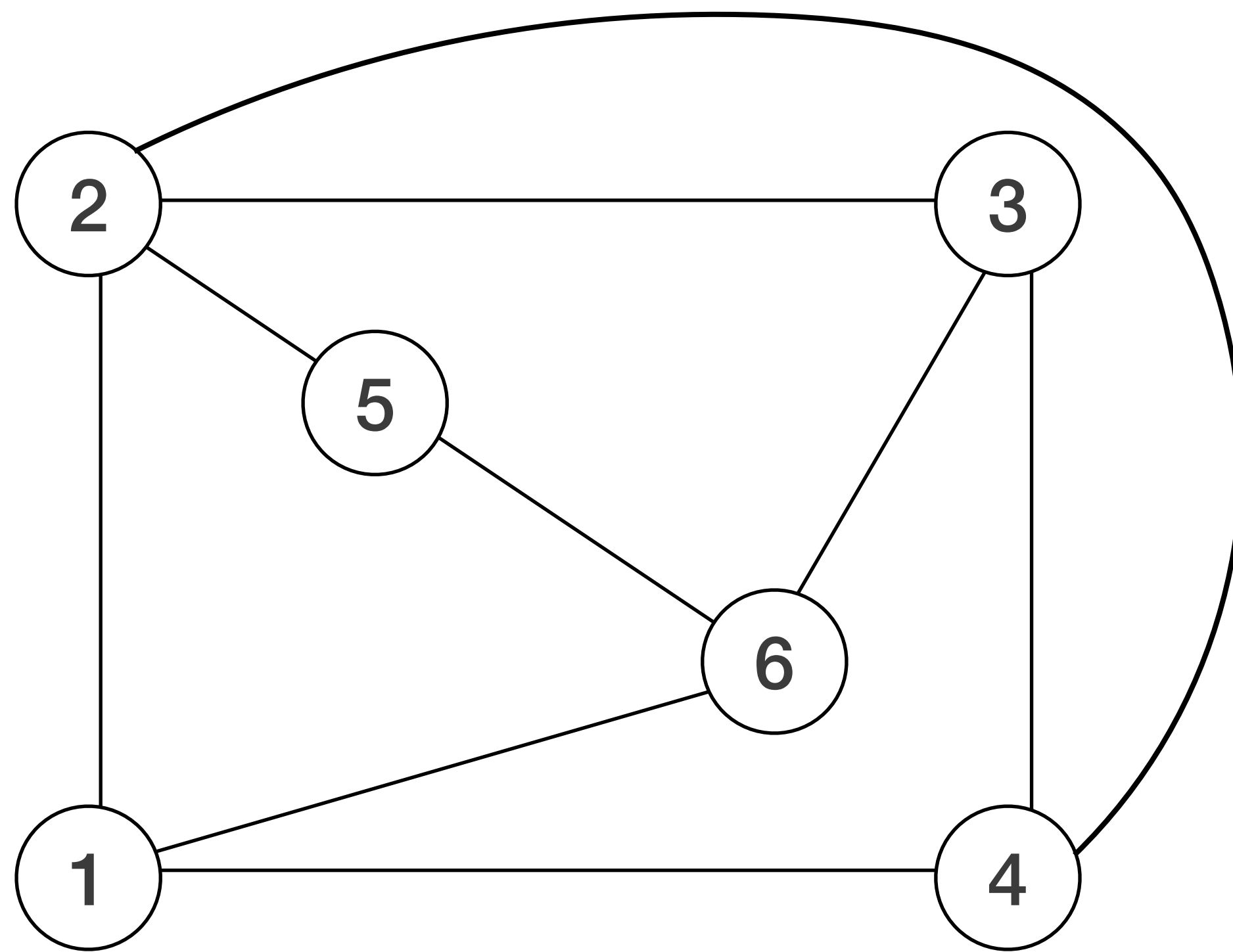
Applications??



What is a dynamic graph ?

- Graph changing with time
- At each time instance — either an edge is ADDED or DELETED

Example:



Time: $t = 4$

1. Incremental scenario:

Only edges are added

2. Decremental scenario:

Only edges are deleted

3. Fully Dynamic scenario:

Edges are added as well as deleted

Goal in Dynamic graph model?

Problems in Static graphs:

MST

Matching

Shortest tree

Connectivity

Dynamic counter part:

This talk:

- Matching
- Connectivity

Main tool:

Randomization

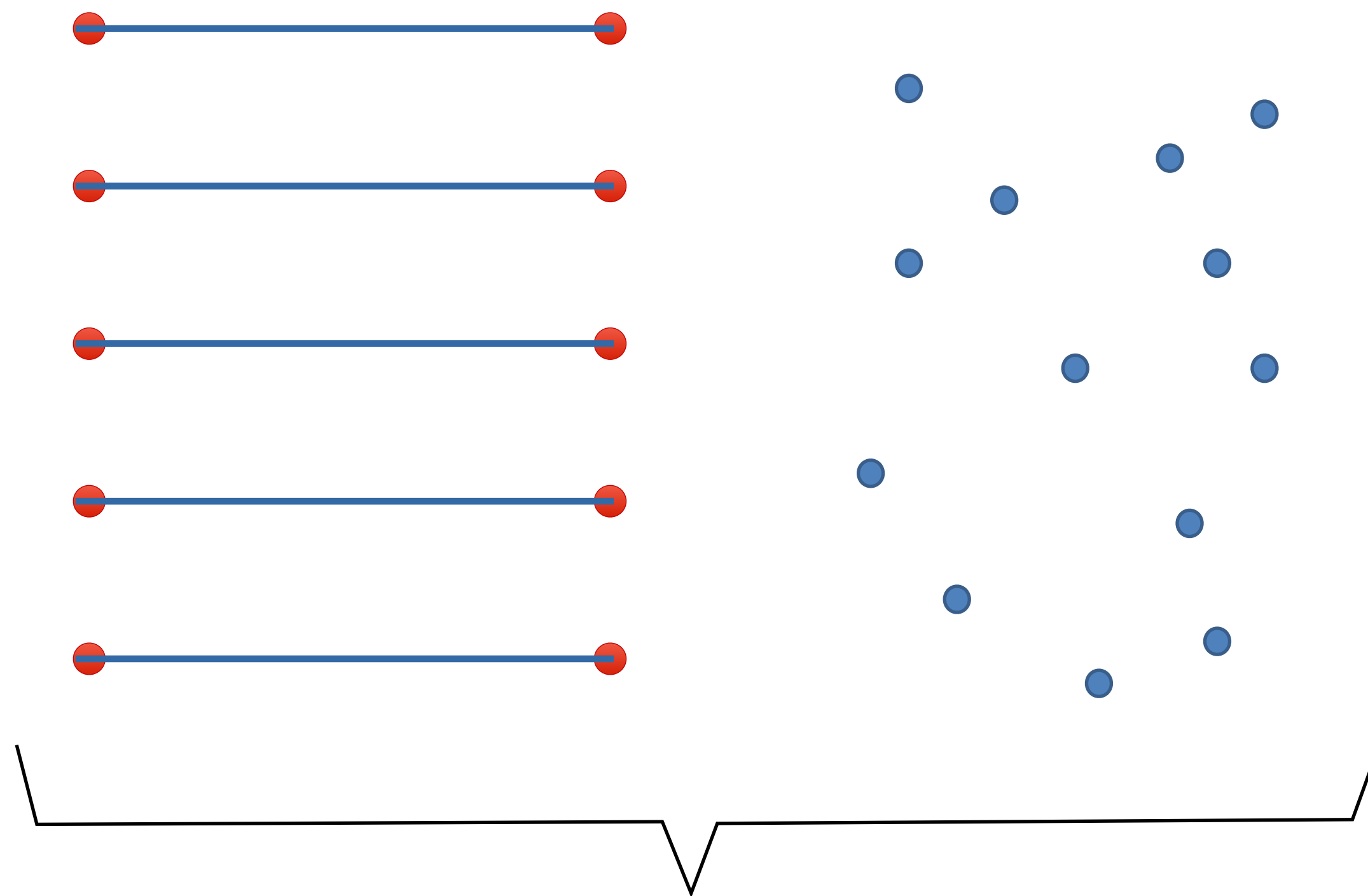
Problem-I: Maximal Matching

Maintain “maximal matching” of a graph

undergoing edge insertions / deletions

Problem-I: Maximal Matching

Maintain “maximal matching” of a graph
undergoing edge insertions / deletions

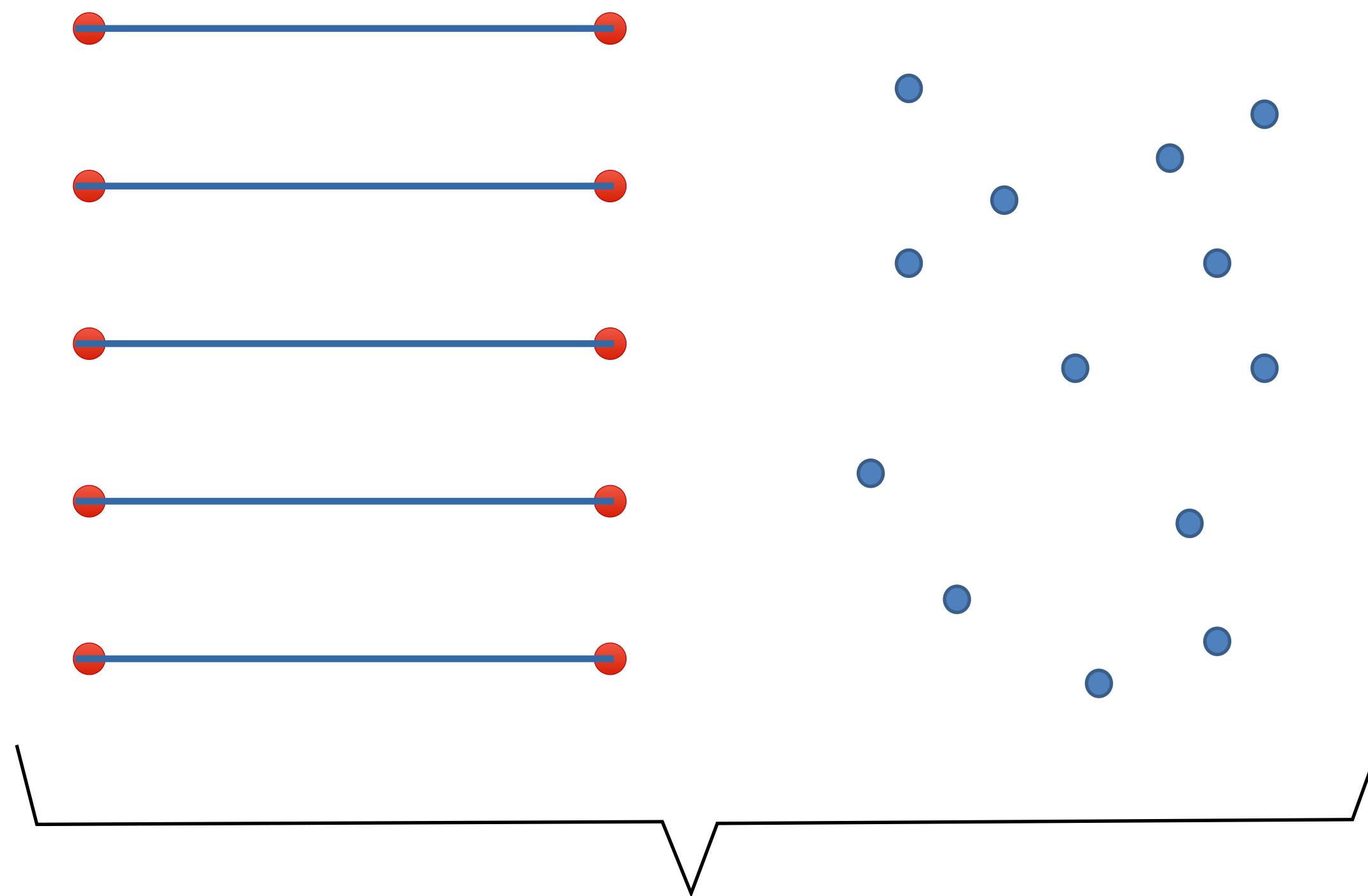


Graph comprising of
matched and unmatched
vertices

- Insertion only: $O(1)$
- Deletion only: $O(n)$
- Fully dynamic: $O(n)$

Problem-I: Maximal Matching

Maintain “maximal matching” of a graph
undergoing edge insertions / deletions



Graph comprising of
matched and unmatched
vertices

**Questions
To Ponder**



Can you dynamically
maintain maximal matching
in sub-linear time?

Problem-II: Connectivity

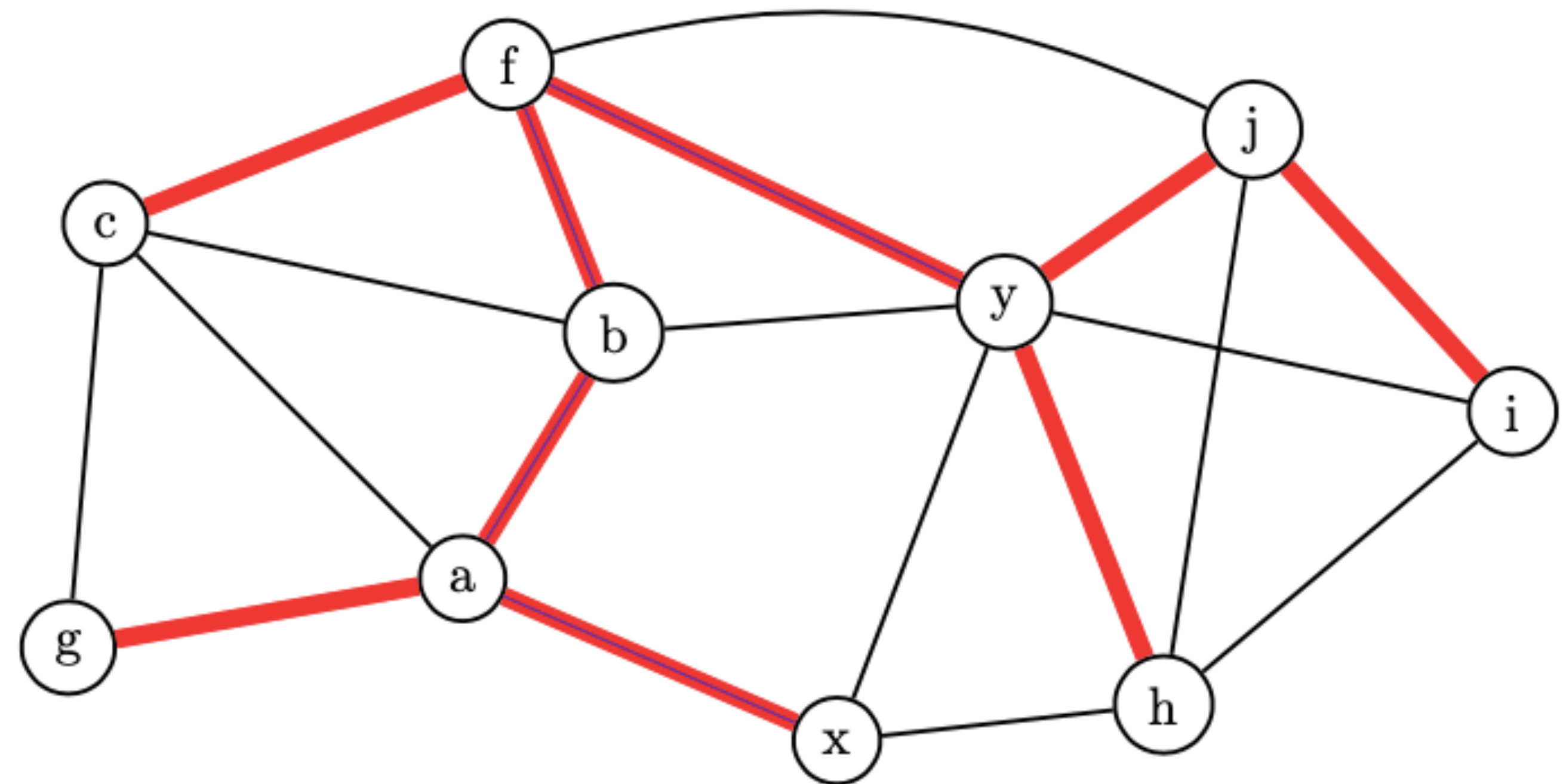
Question : What is a certificate for connectivity?

Problem-II: Connectivity

Question : What is a certificate for connectivity?

ALGORITHM
(for static case):

1. Let $(e_1, \dots, e_m)$ be edges of G
2. $F = (V, \emptyset)$
3. For $i = 1$ to m :
 **If (endpoints of e_i belong to
 different components of F):**
 Add e_i to F
4. Return F



Problem-II: Connectivity (Spanning forest)

Incremental:

On addition of edge e_i we check if endpoints of e_i belong to different components of current F .

If yes, then add e_i to F

Questions
To Ponder

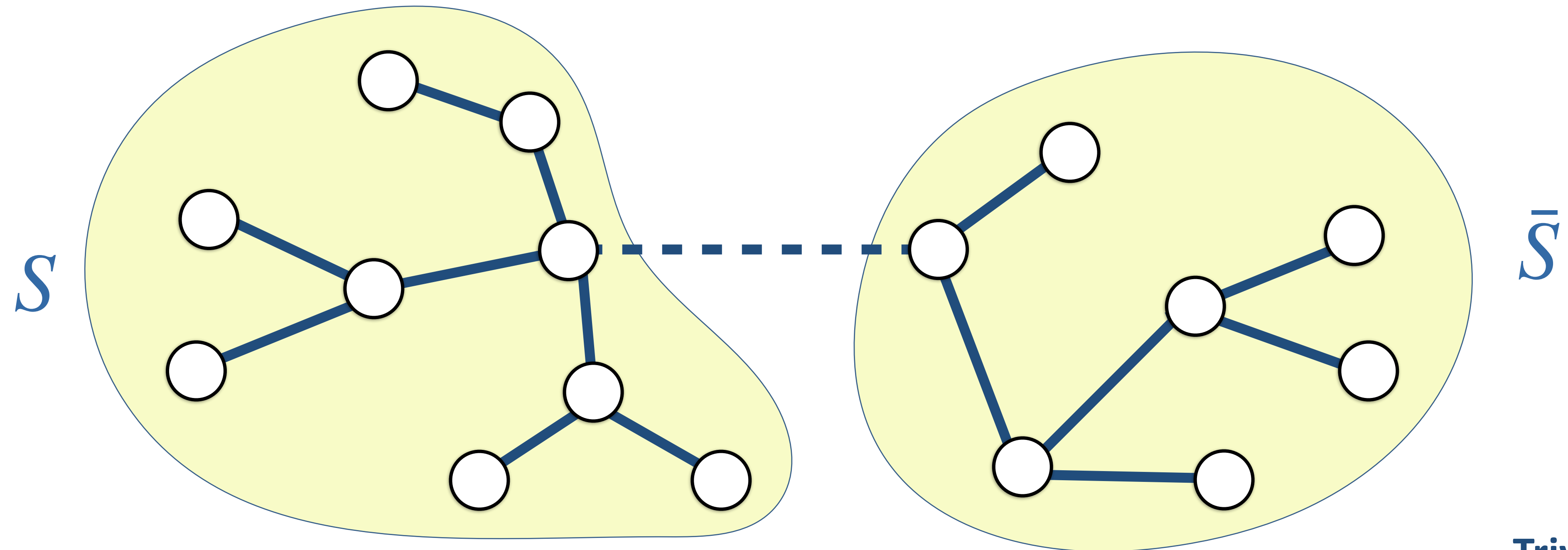


Can you incrementally
maintain spanning forest in
 $o(n)$ worst-case update
time?

Problem-II: Connectivity (Spanning forest)

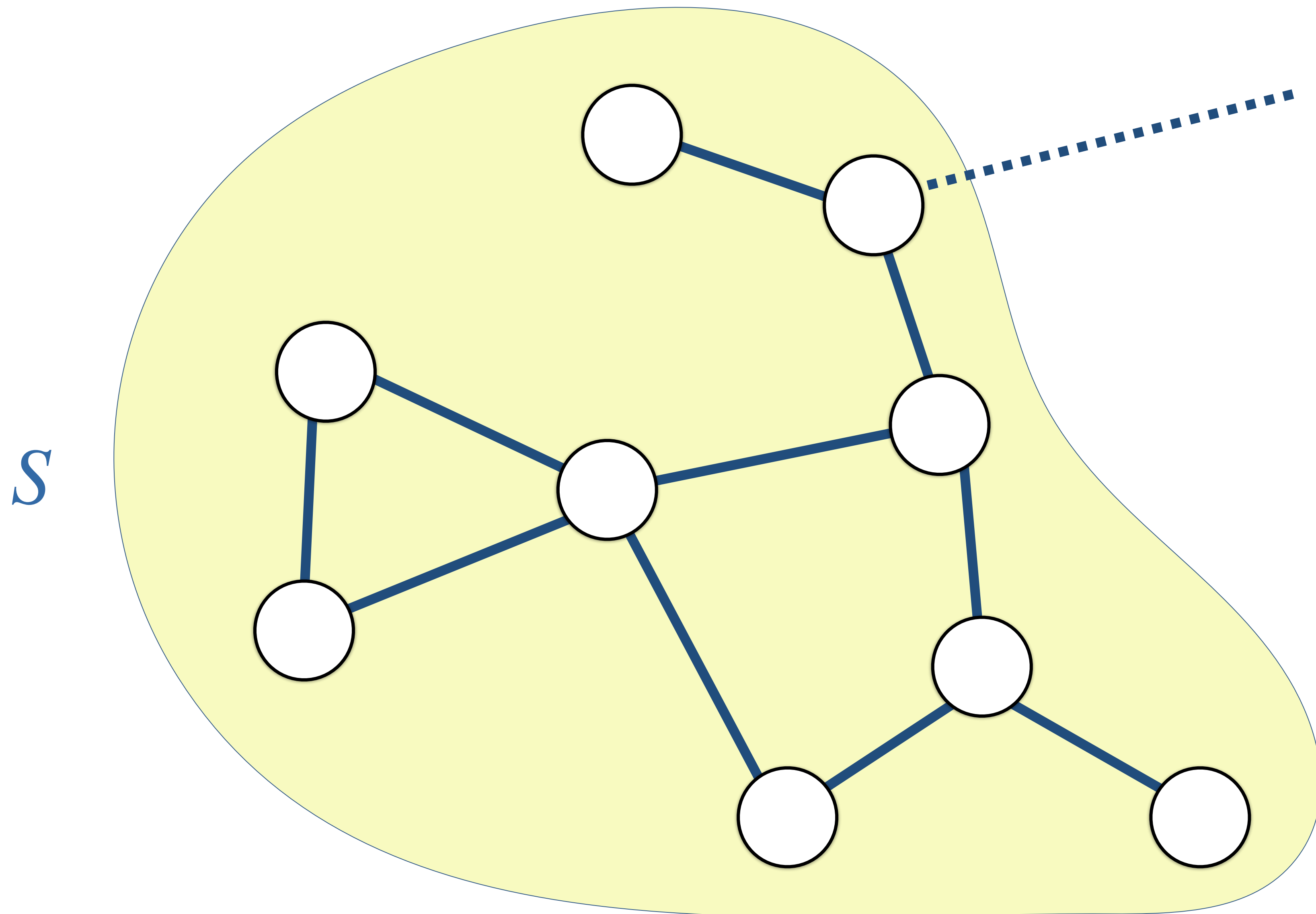
Question: How to find an edge going out from a set S after deletion?

Decremental:

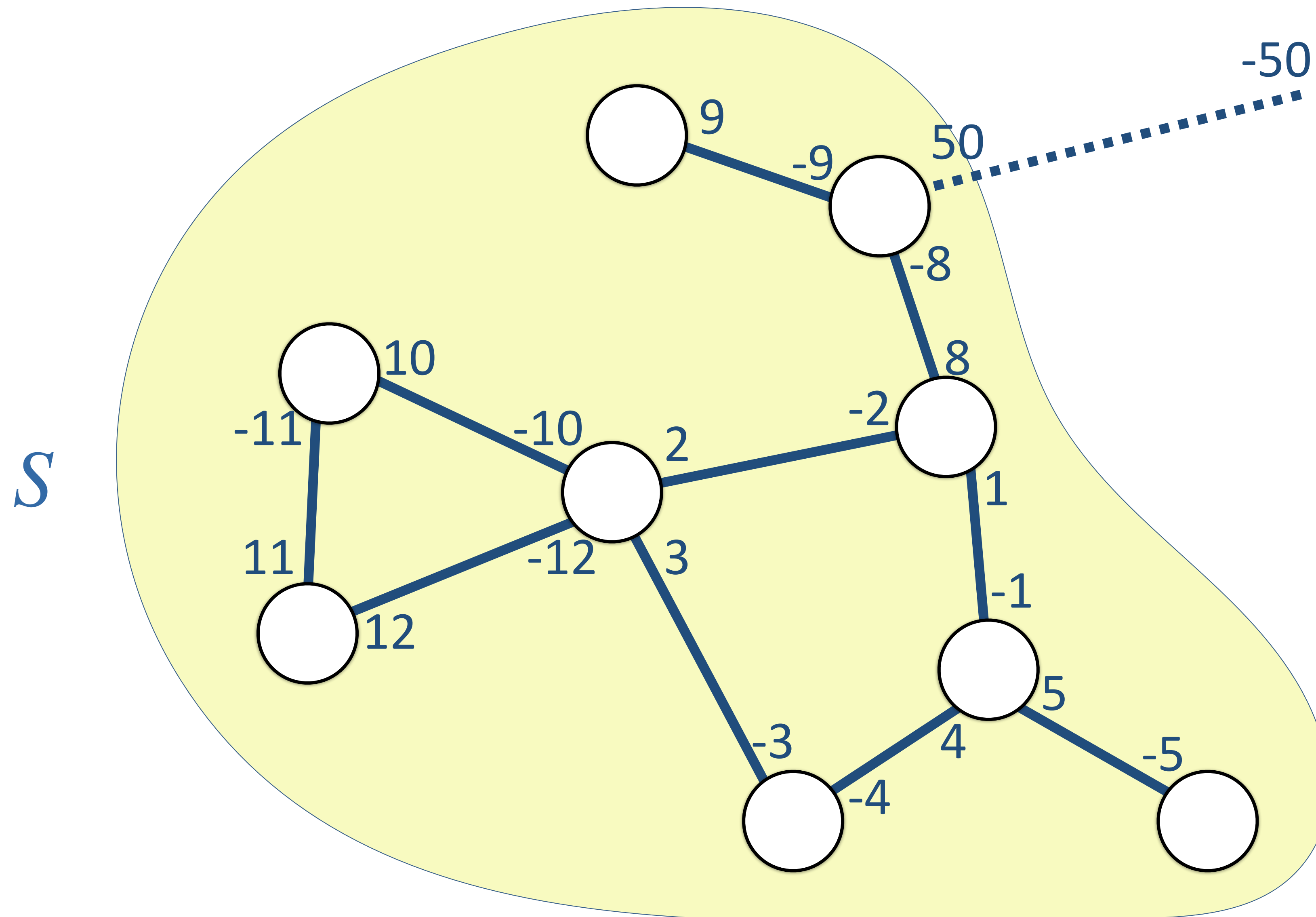


Trivial solution
takes $O(n^2)$
time!

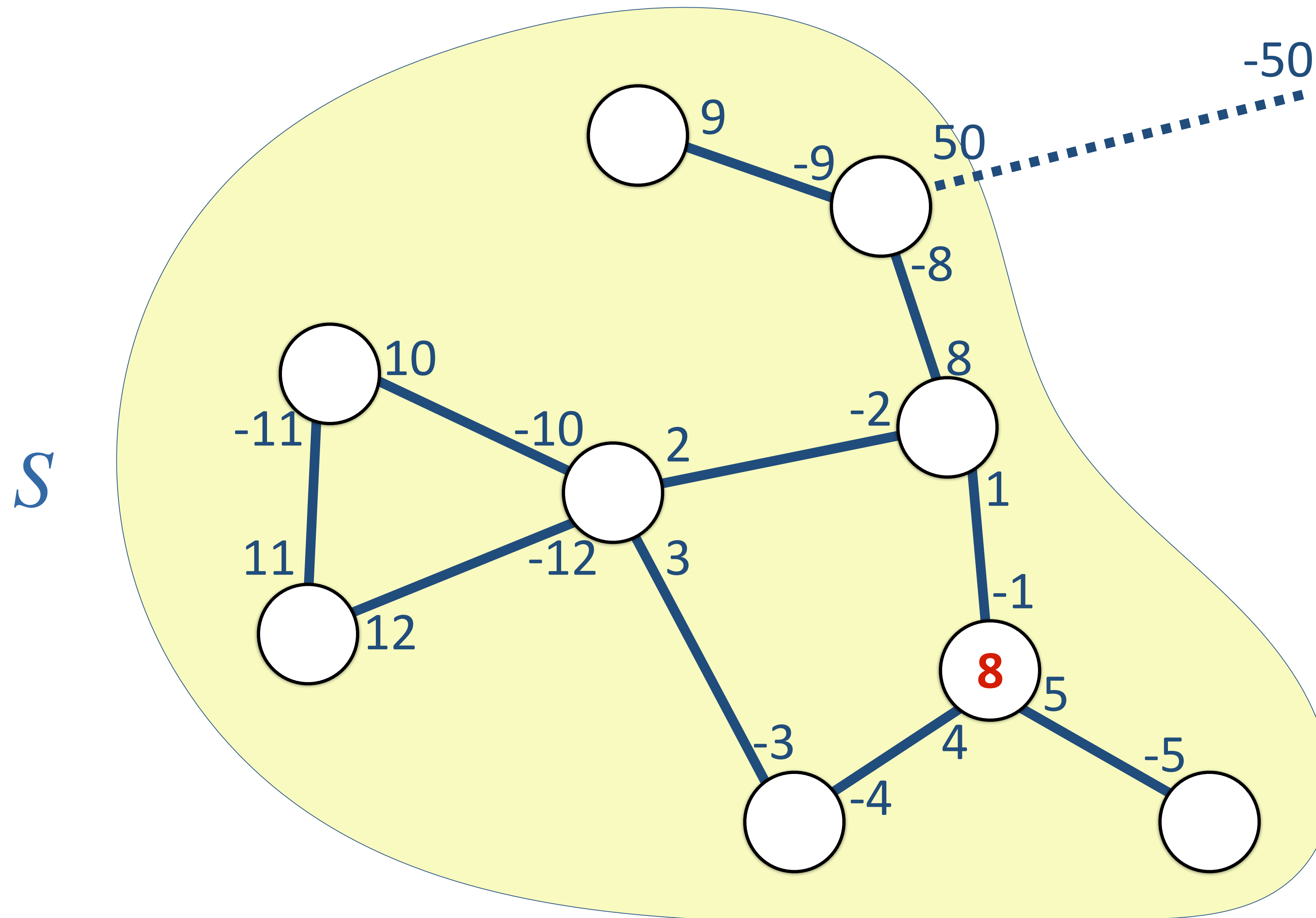
Simpler Question: What if there is only one edge going out from a set S in updated graph?



Simpler Question: What if there is only one edge going out from a set S in updated graph?



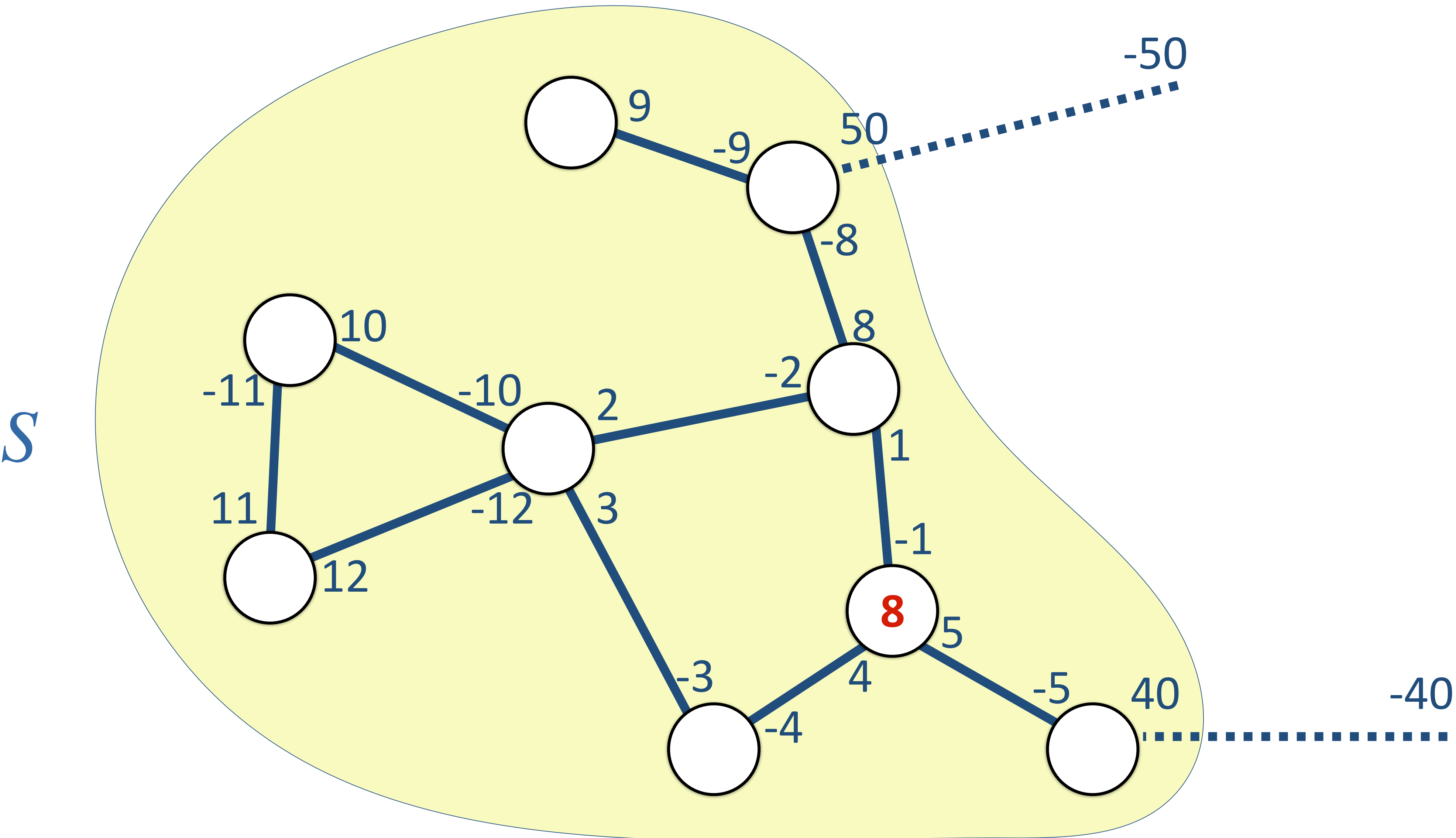
Simpler Question: What if there is only one edge going out from a set S in updated graph?



$\alpha(w)$ = Sum of signed edge labels incident to w

QUESTION:
Can you find the
unique edge
going out from S
in $O(|S|)$ time?

Next Question: What if there are TWO edges going out from a set S in updated graph?



Puzzle: Guess a set with exactly one coin

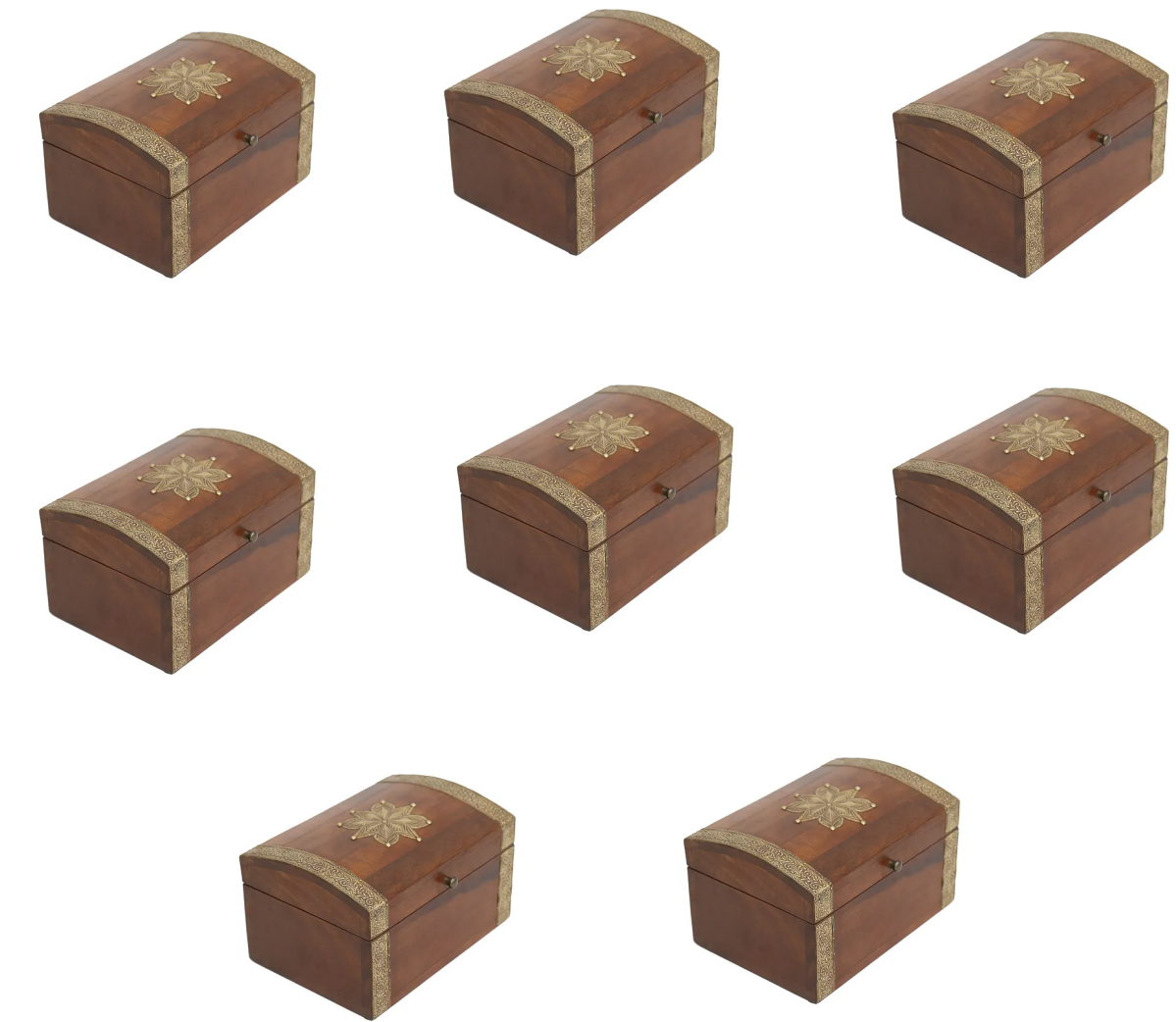
- n Boxes
- Some k Boxes have a Coin.
- You **win** a round if you guess a set S of boxes such that exactly one box in this set has a coin.

Question: How many rounds are needed to win at least one game?



Puzzle: Guess a set with exactly one coin

- n Boxes
- Some k Boxes have a Coin.
- You **win** a round if you guess a set S of boxes such that exactly one box in this set has a coin.



Question: How many rounds are needed to win at least one game?

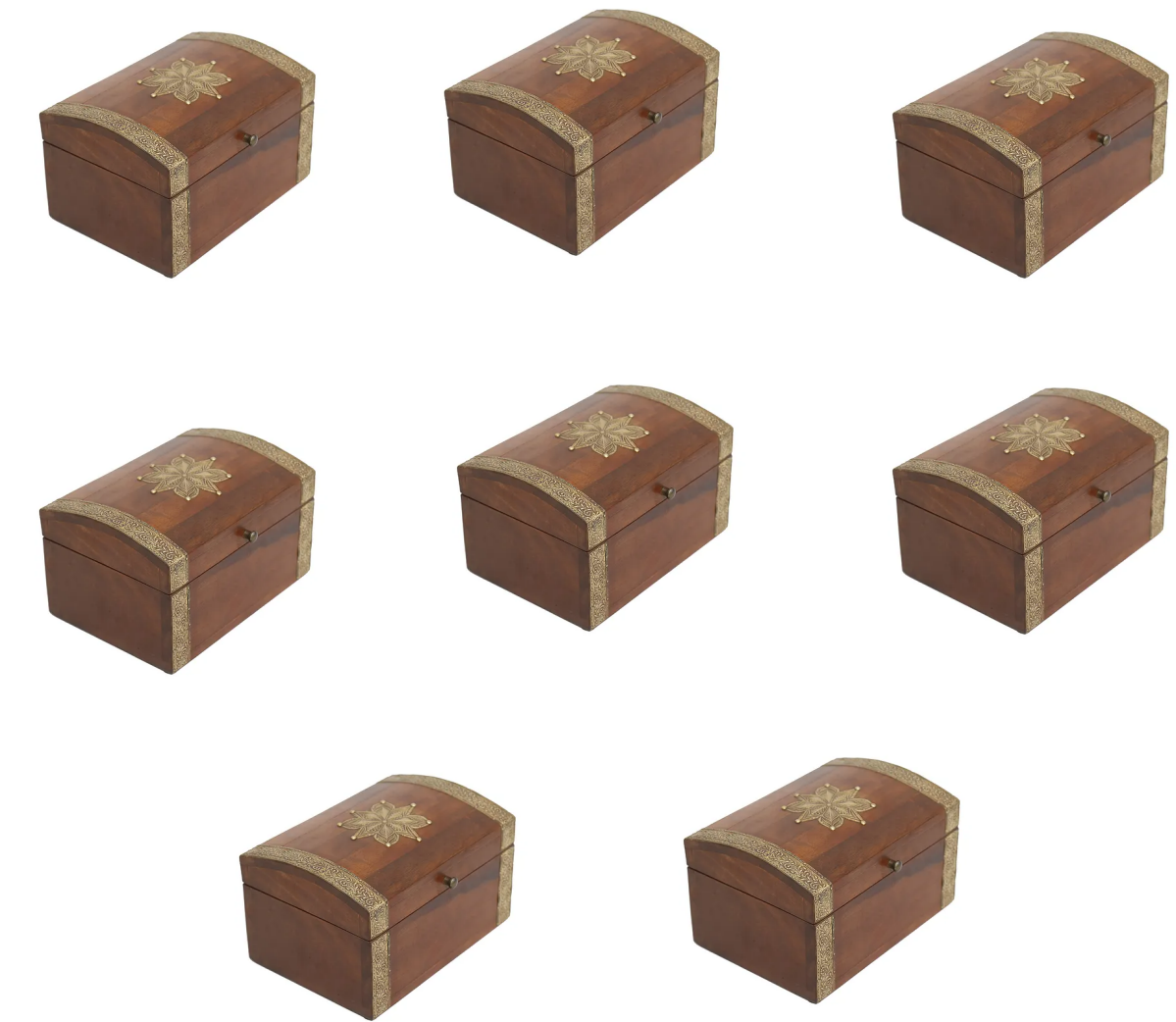
Strategy when k is known:

Compute a set by choosing each box with prob $\frac{1}{k}$, then success prob is ${}^kC_1\left(\frac{1}{k}\right)\left(1 - \frac{1}{k}\right)^{k-1} \geq \frac{1}{4}$.

To boost success probability - Repeat $O(\log n)$ times.

Puzzle: Guess a set with exactly one coin

- n Boxes
- Some k Boxes have a Coin.
- You **win** a round if you guess a set S of boxes such that exactly one box in this set has a coin.



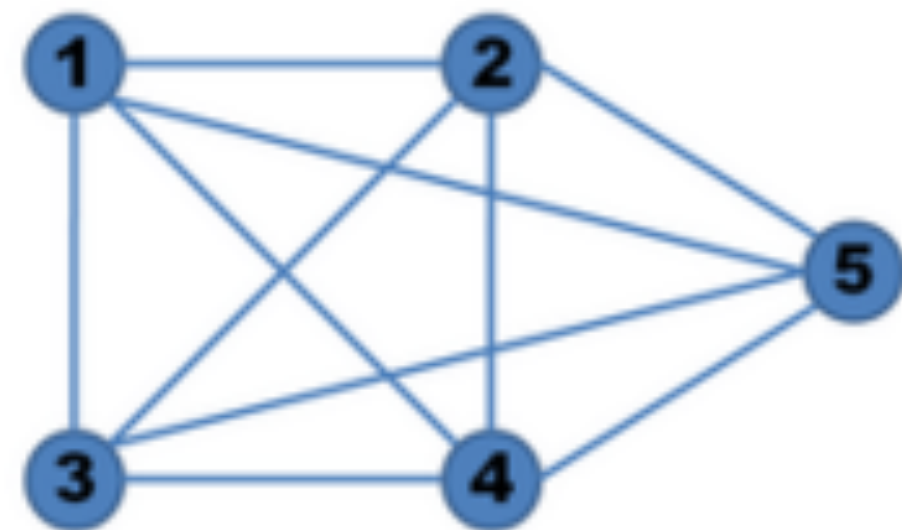
Question: How many rounds are needed to win at least one game?

Strategy when k is unknown: Suppose we have a z satisfying $k \in [z, 2z]$.

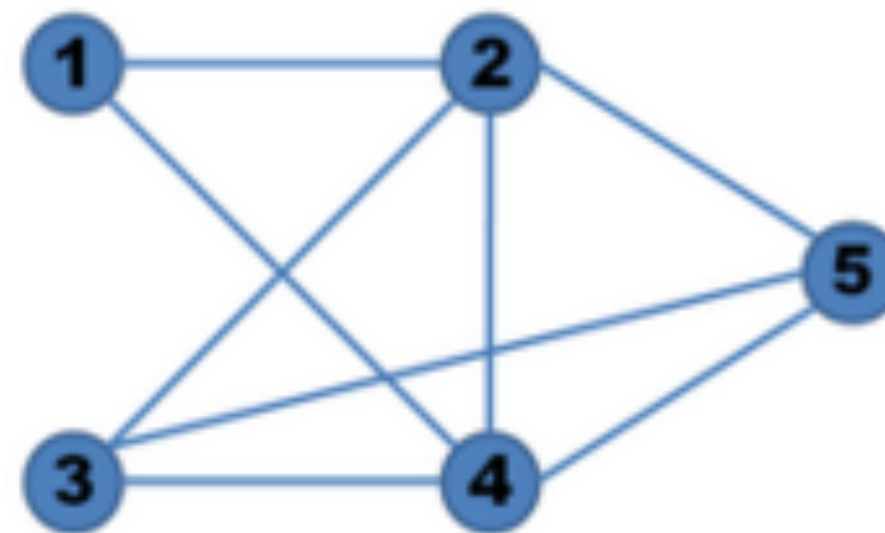
Compute a set by choosing each box with prob $\frac{1}{z}$, then success prob is ${}^kC_1\left(\frac{1}{z}\right)\left(1 - \frac{1}{z}\right)^{k-1} \geq \frac{1}{16}$.

To boost success probability - Repeat $O(\log n)$ times.

Family of Subgraphs



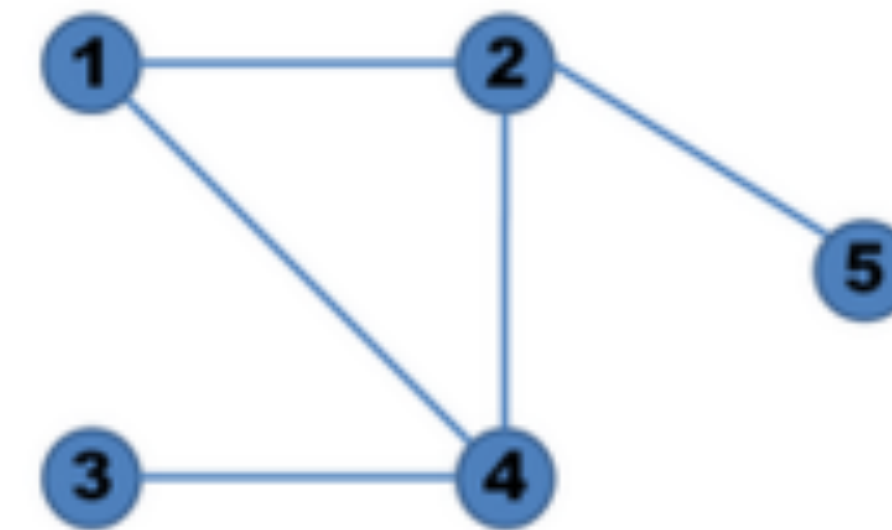
$H_0 \leftarrow G$



H_1 - each edge of G

with probability $\frac{1}{2}$

lies in H_1



H_2 - each edge of G

with probability $\frac{1}{4}$

lies in H_2

...

Graph family — $H_0, H_1, \dots, H_{\log_2 n}$

- To boost success probability - Repeat construction of entire graph family $O(\log n)$ times.

Dynamic-Connectivity-Graph-Structure

Preprocessing

1. Initialize F as the spanning forest of the current graph.
2. Compute $O(\log^2 n)$ independent subgraphs $H_{i,r}$ of G such that for each $i, r \in [1, 8 \log_2 n]$, $H_{i,r}$ contains each edge of G with probability $1/2^i$.
3. Compute $\alpha(v | H_{i,r})$ for each node v , and for each $i, r \in [1, 8 \log_2 n]$.

Addition of edge $e = (x, y)$

1. For each $i, r \in [1, 8 \log_2 n]$:
 - Add edge (x, y) to subgraph $H_{i,r}$ with probability $1/2^i$.
 - Update $\alpha(x | H_{i,r})$ and $\alpha(y | H_{i,r})$.
2. If x, y lie in different components of F , then add edge to forest F .

Deletion of edge $e = (x, y)$

1. For each $i, r \in [1, 8 \log_2 n]$:
 - Update $\alpha(x | H_{i,r})$ and $\alpha(y | H_{i,r})$.
2. If (x, y) was part of F :
 - Remove edge (x, y) from F , and compute the resultant components $(S, \bar{S})$.
 - Use subgraphs $H_{i,r}$ to find a replacement edge connecting $S, \bar{S}$ (if it exists), and add it to forest F .

Dynamic-Connectivity-Graph-Structure

We can dynamically maintain spanning forest of an n vertex graph undergoing edge insertions/deletions in $O(n \log^2 n)$ update time. The algorithm uses randomization and has success probability $1 - \frac{1}{n^4}$.



Can you use AVL trees to reduce the update time to $O(\text{polylog}(n))$?

Thank you!