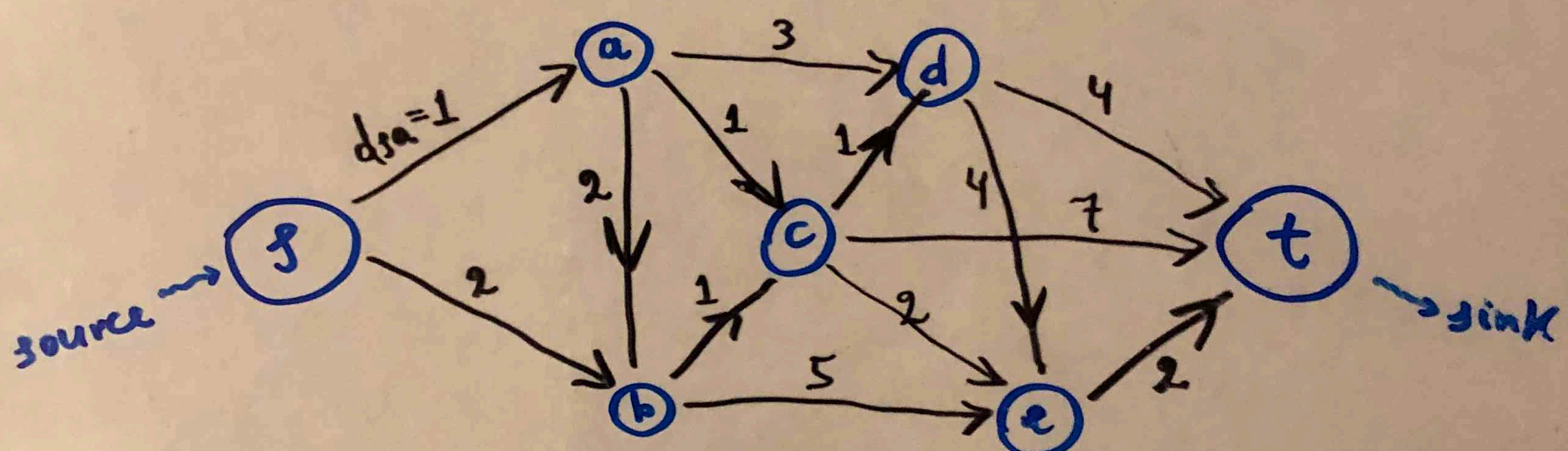
Special cases of Min-Cost Flow Problem (cont.)

Shortest Path Problem

- one source & one sink
- distances on arcs
- Find the shortest path from source to sink



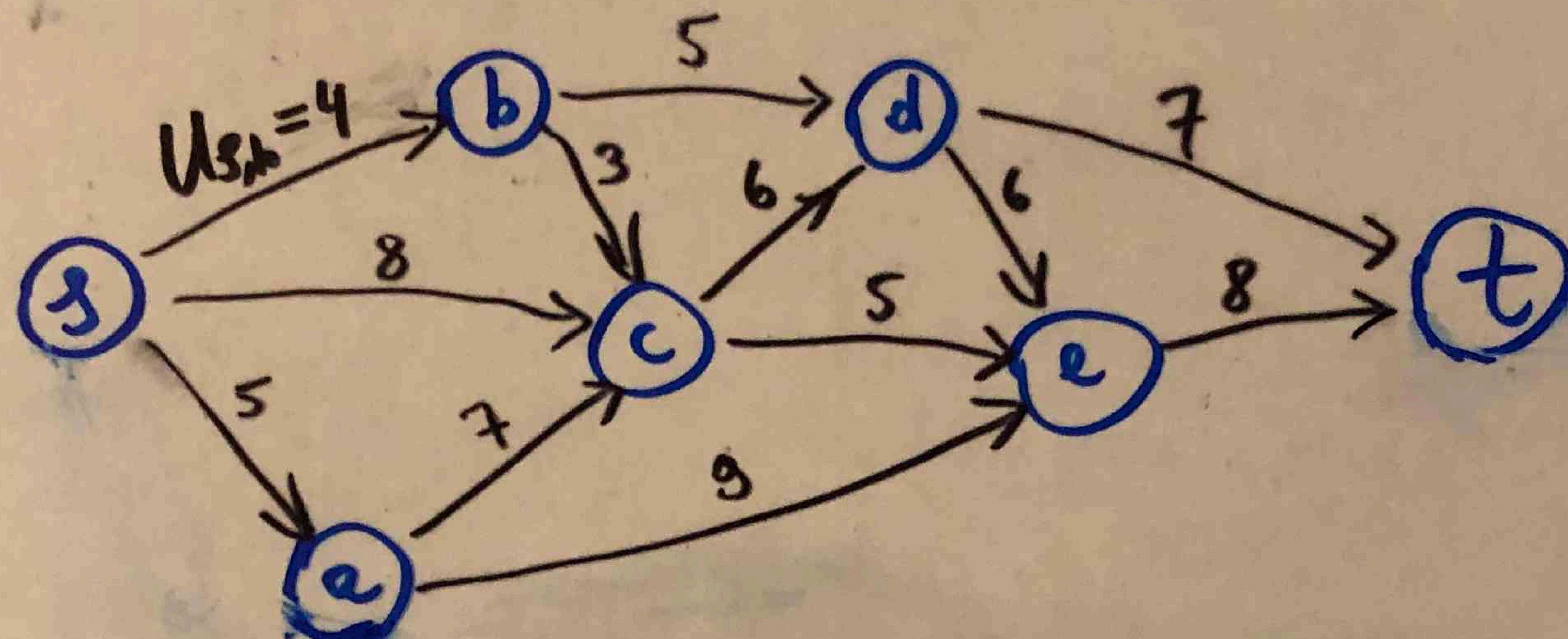
Q: How to formulate this as a min-cost flow problem.

- Assign $b_s=1$, $b_t=-1$, $b_i=0$ for all other nodes i
- costs = distances
- no upper bounds necessary

Note: • This formulation gives a reasonable solution to Shortest Path problem based on Integrality Property.

Maximum Flow Problem

- source s and sink t
- upper bound u_{ij} on each arc $i \to j$
- **Goal:** Send as much flow as possible from s to t.



Formulating a Min-Cost Flow Problem

- Set $c_{ij}=0$ for all existing arcs $i \to j$
- Set $b_s = F$, $b_t = -F$ where F is a safe upper bound for max flow

  $b_i = 0$ for other nodes.
- Add an arc $s \to t$ with $c_{s,t} = M$, $u_{s,t} = +\infty$ ($M \gg 0$ is a large #)
- Since $c_{s,t} = M$, the Min-cost Flow Problem will send as much as possible by real arcs and the rest of the demand by artificial arc $s \to t$.

Note: Integrality property will provide a reasonable (integer) sol-n to the problem

All the special cases we discussed,

Transportation Problem
Assignment Problem
Shortest Path Problem
Max Flow Problem

can be solved in two ways:

1) formulating as Min-Cost Flow and applying a Min-Cost Flow algorithm (e.g., Network Simplex)

2) Applying special-purpose algorithms designed for each of those cases

---

An example of a Network Model which is <u>not</u> a special case of Min-Cost Flow:

Min Spanning Tree Problem

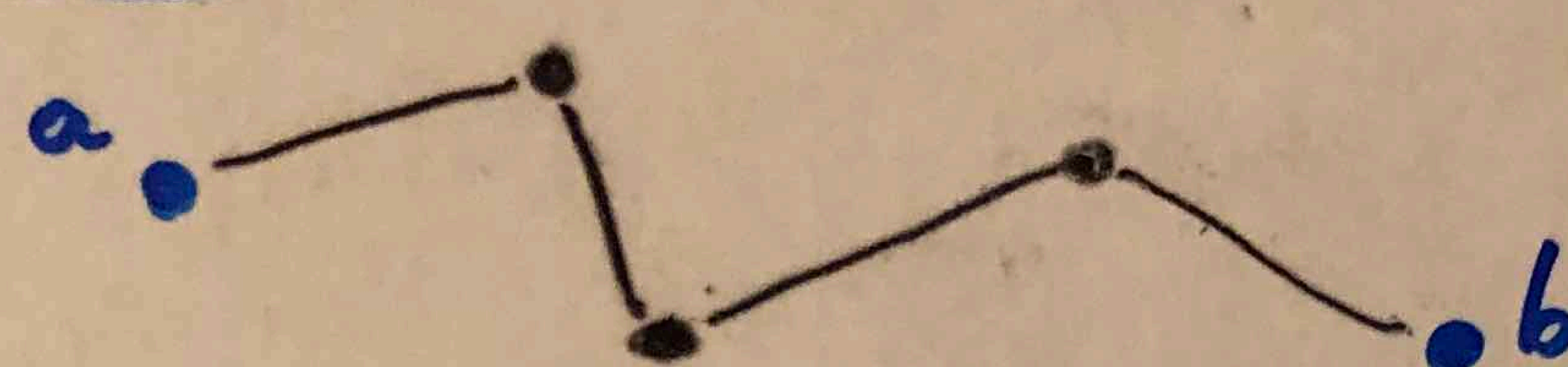
Given • undirected graph $G=(V,E)$, $|V|=n$
• cost function: $c: E \to \mathbb{R}$

Find min-cost spanning tree for V.

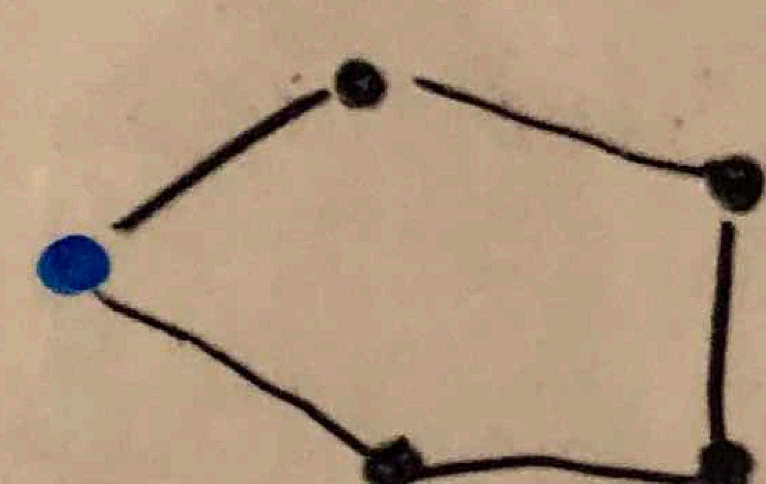
That is, find subset of arcs $E' \subseteq E$ which connects any two nodes of V with minimal possible cost.

Some Terminology of Networks
(before going to Min Span.. Tree problem)

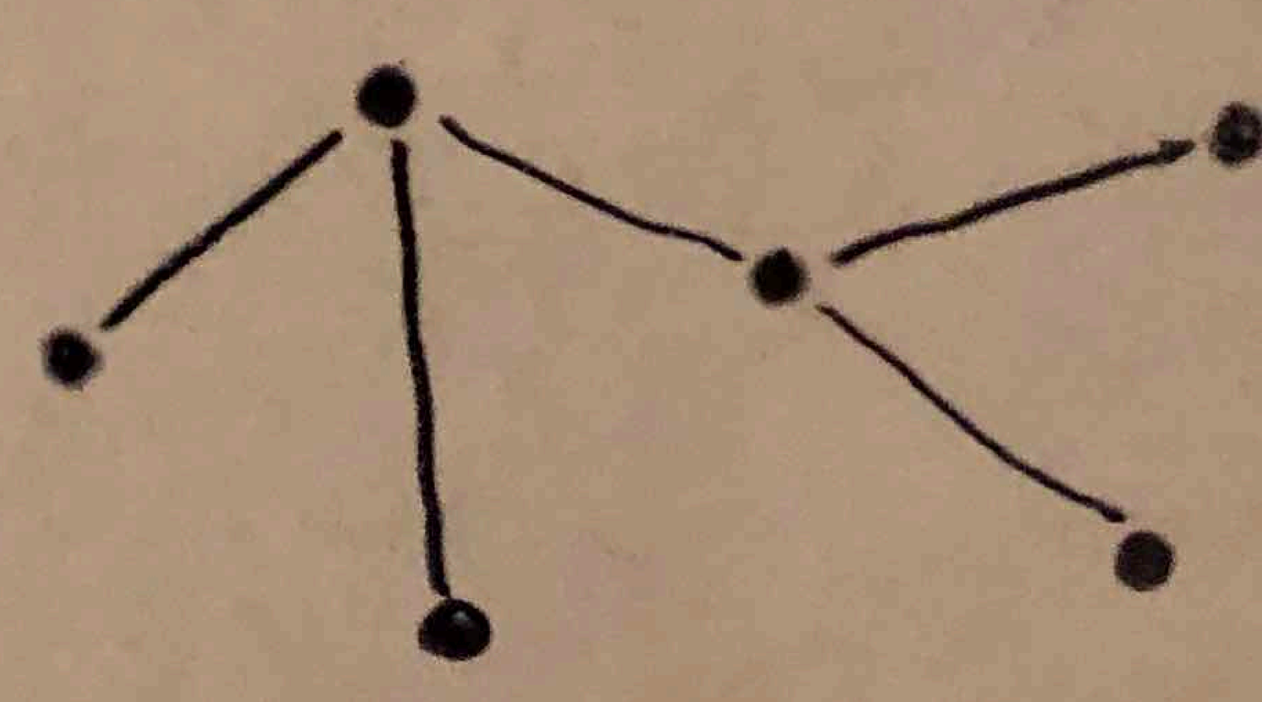
- A path between two nodes is a sequence of distinct arcs connecting these nodes.



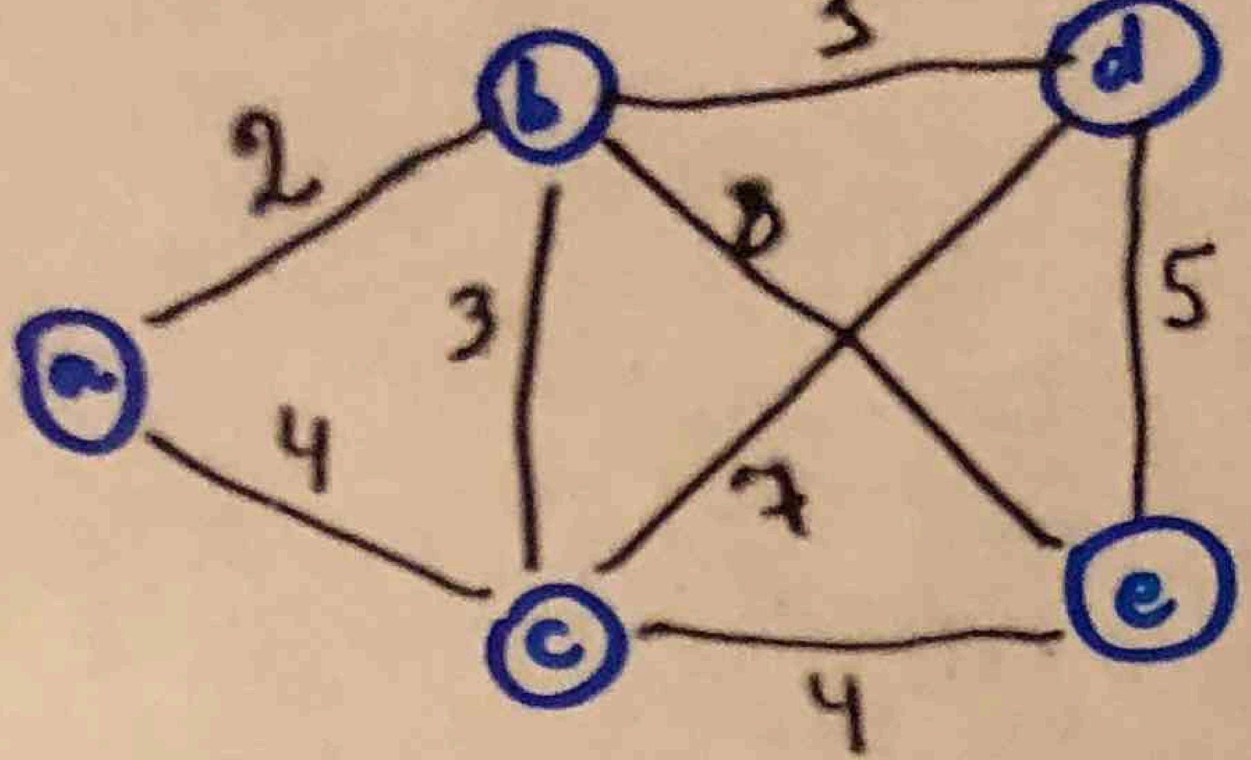
- A path that begins and ends at the same node is called a cycle.



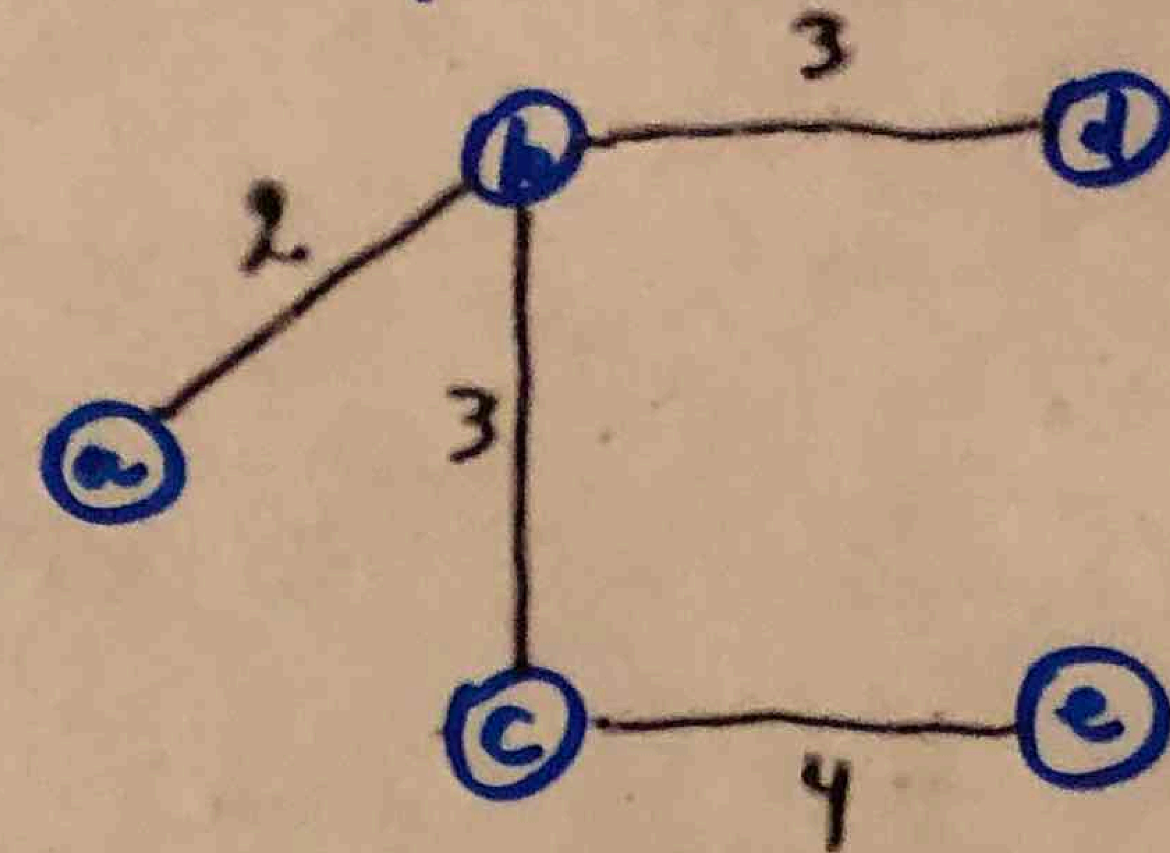
- Two nodes are connected if there is a path between them.
- A graph is connected if every pair of nodes is connected.
- A graph is acyclic if it doesn't have any cycle.
- A graph is called tree if it is connected and acyclic.



Example: G=(V,E)



min-spanning tree: G=(V,E)

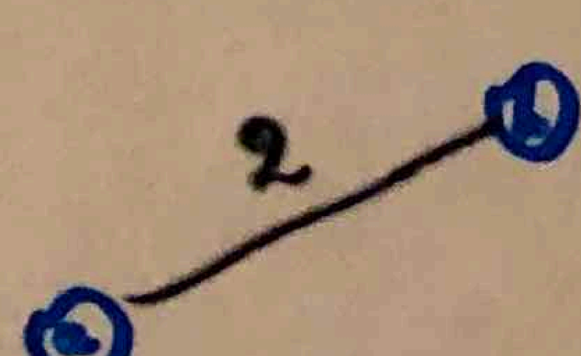


Algorithm for solving Min Span. Tree Problem

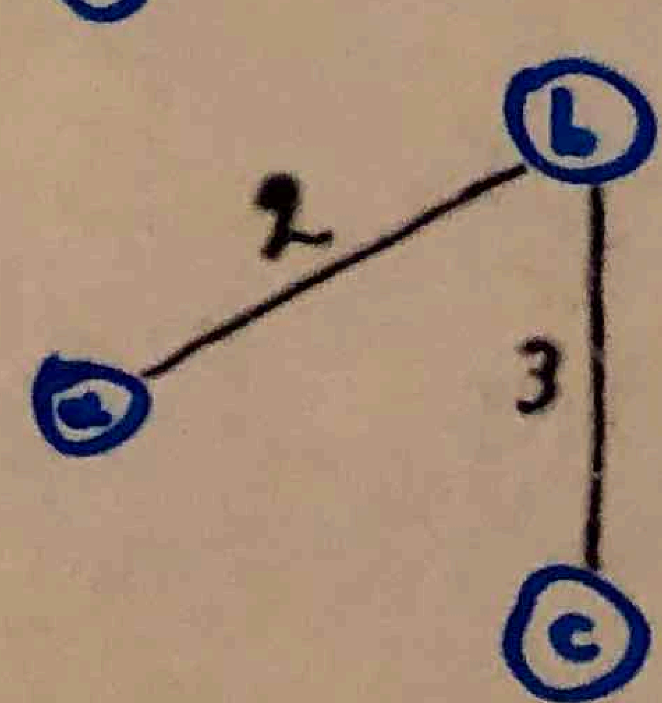
- Select any node arbitrarily, connect to its nearest node.
- REPEAT
  Identify the unconnected node which is closest to a connected node; Connect those 2 nodes.
  until all nodes are connected.

Ex. (cont.)

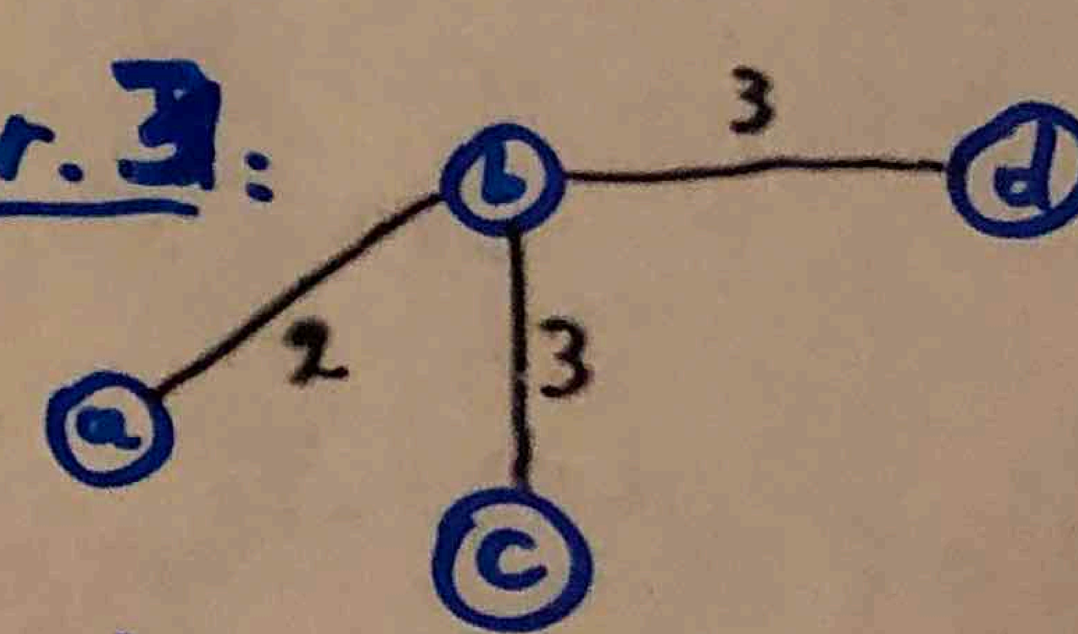
- Start with node a
- Iter. 1:



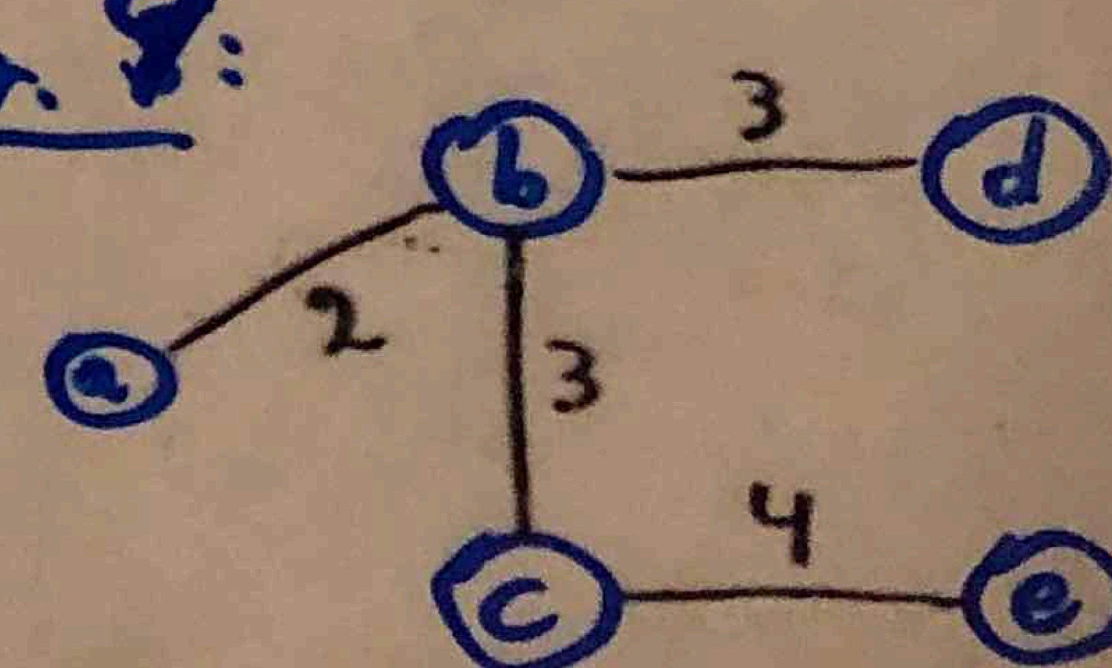
- Iter. 2:



- Iter. 3:



- Iter. 4:



Note: If $|V| = n$ then spanning tree of V has $n-1$ arcs.

Now back to

Min Cost Flow Problem

Recall LP formulation:

$$\min \sum_{i \to j \in E} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_{k:(j,k) \in E} x_{j,k} - \sum_{i:(i,j) \in E} x_{ij} = b_j, \quad \forall \text{node } j$$

(cons.-of-flow)

$$x_{ij} \geq 0, \quad \forall \text{arc } i \to j$$

(we will use this simplified version of bound constraints)

- Want to give a streamlined version of Simplex for solving this specific LP. It is called Network Simplex.

- Correspondence between basic solutions and spanning trees
    - # of constraints $= n$ but one of the constraints is redundant $\Rightarrow$

\# of nonredundant constraints is $n-1$

$\Rightarrow$ # of basic variables is $n-1$

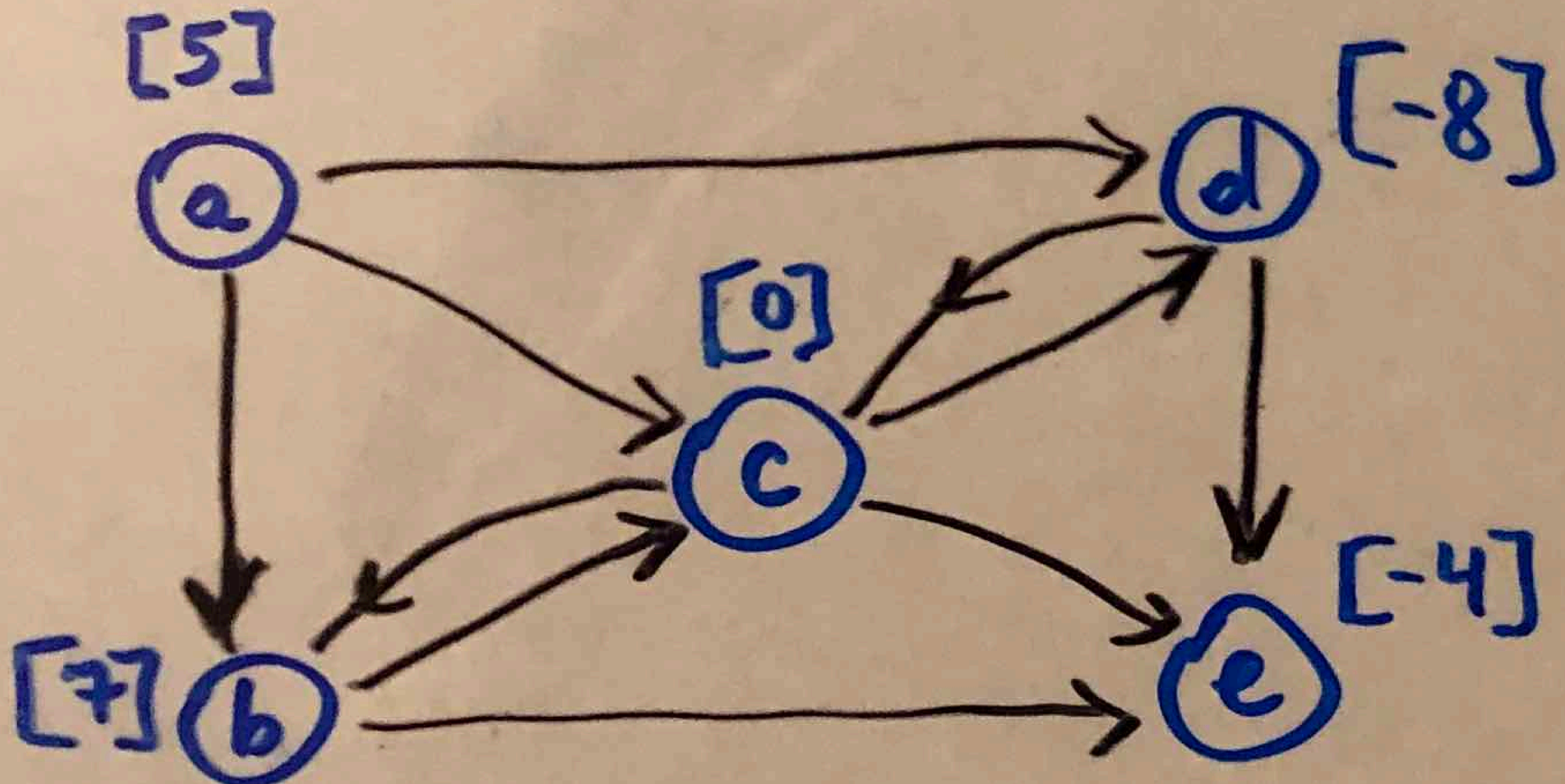
$\Rightarrow$ in any basic sol-n,
only $n-1$ arcs have nonzero values

- basic arcs never form (undirected) cycles.

- $\left.\begin{array}{l} n-1 \text{ basic arcs;} \\ \text{no cycles} \end{array}\right\} \Rightarrow$ basic arcs form a spanning tree for V

- Thus, for Min Cost Flow Problem
  basic solutions $\Leftrightarrow$ spanning tree solutions

Example:



Basic sol-n: