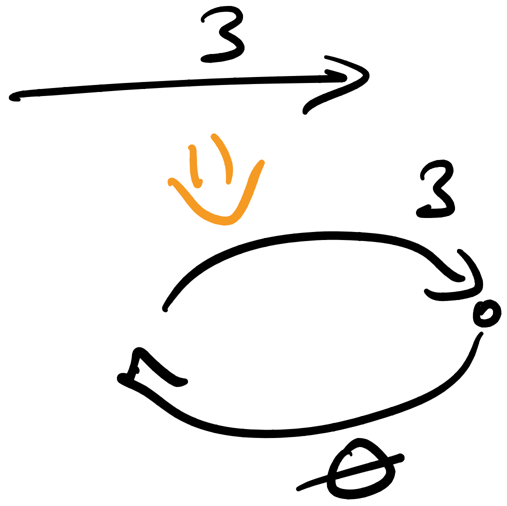
Network flow algorithms

Ibrahim Numanagić

Introduction

- Input: a directed graph where each edge is a pipe with some capacity
- How much flow you can go from the source to the sink?



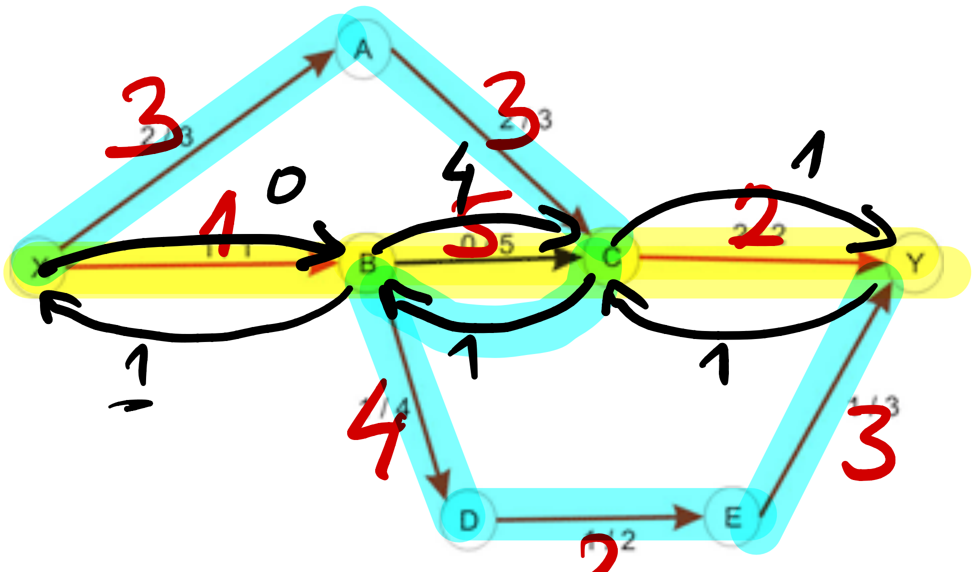
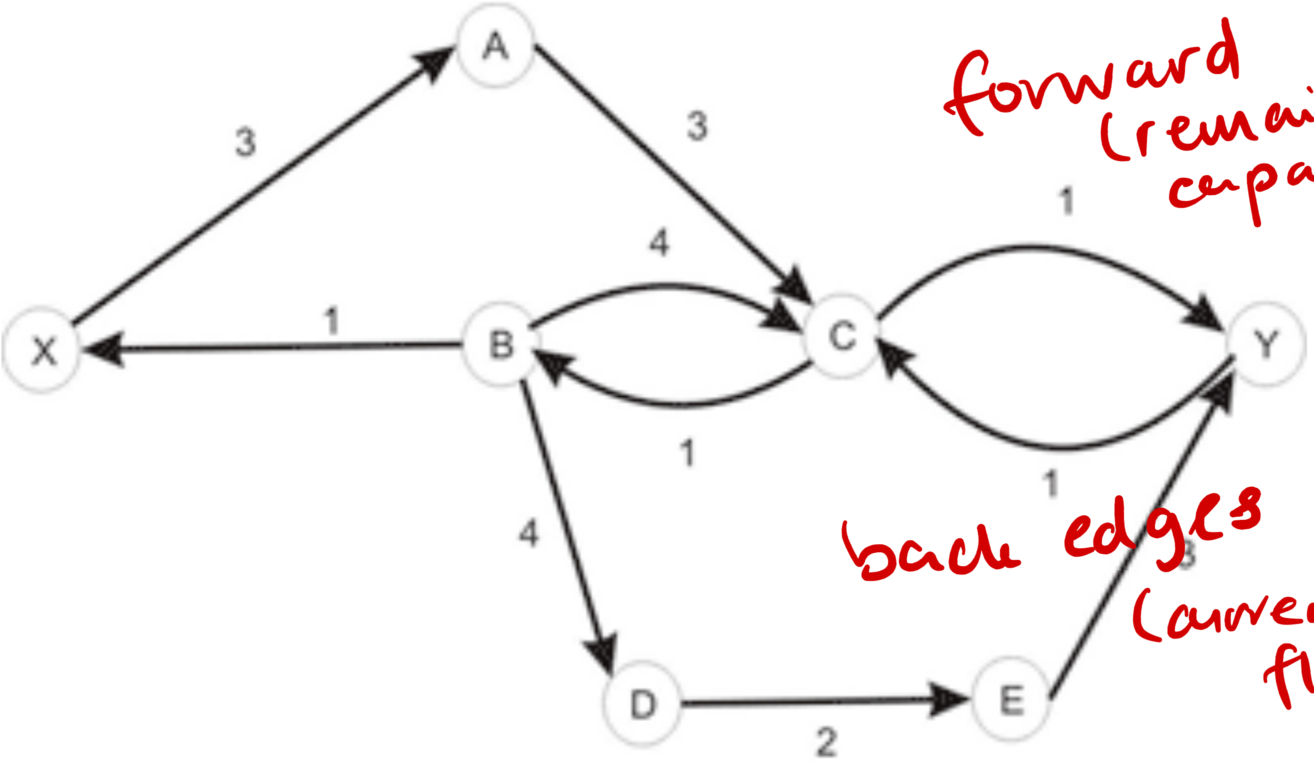


Figure 1a - Maximum Flow in a network

Concepts

- Residual network
- Augmenting paths

paths in residual graph from source to sink



forward (remaining capacity)

back edges (current flow)

Figure 2b - The residual network of the network in 2a

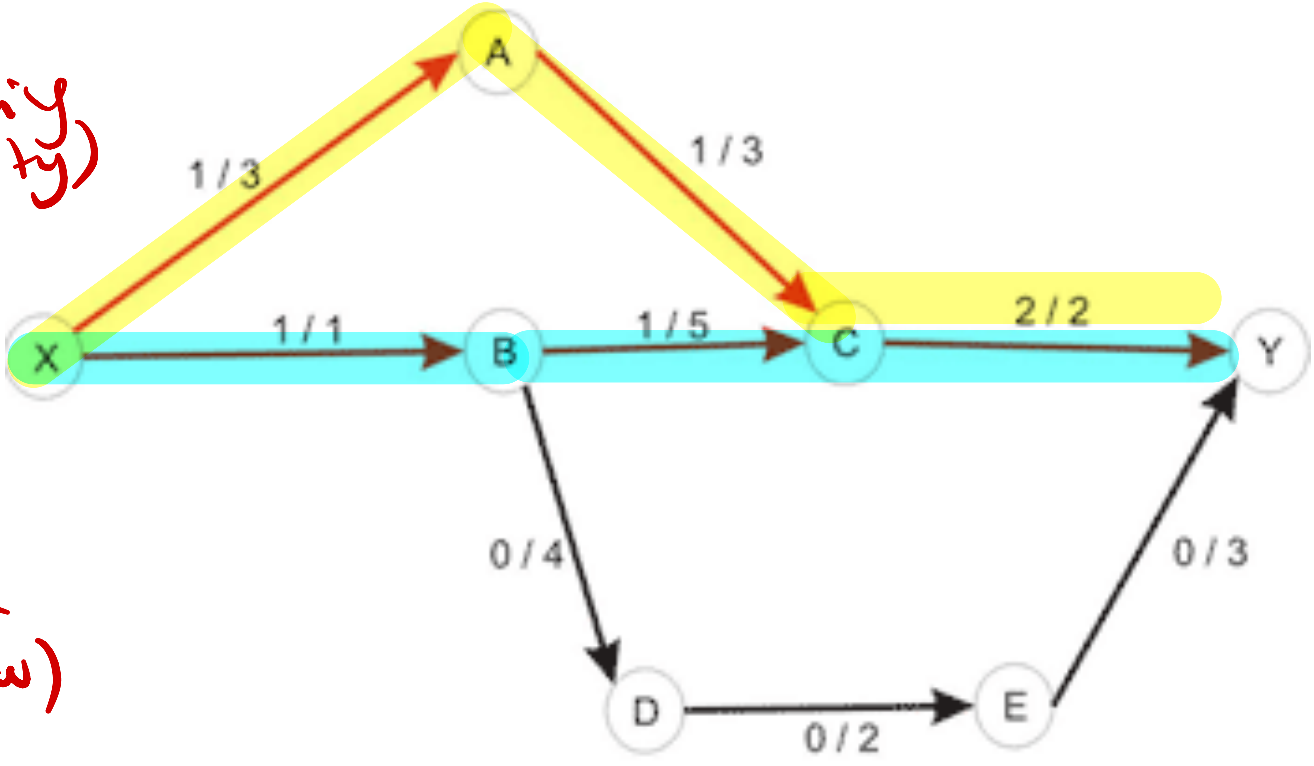


Figure 3a

residual network

Ford-Fulkerson algorithm

- Keep finding augmenting paths in the residual network while they exist

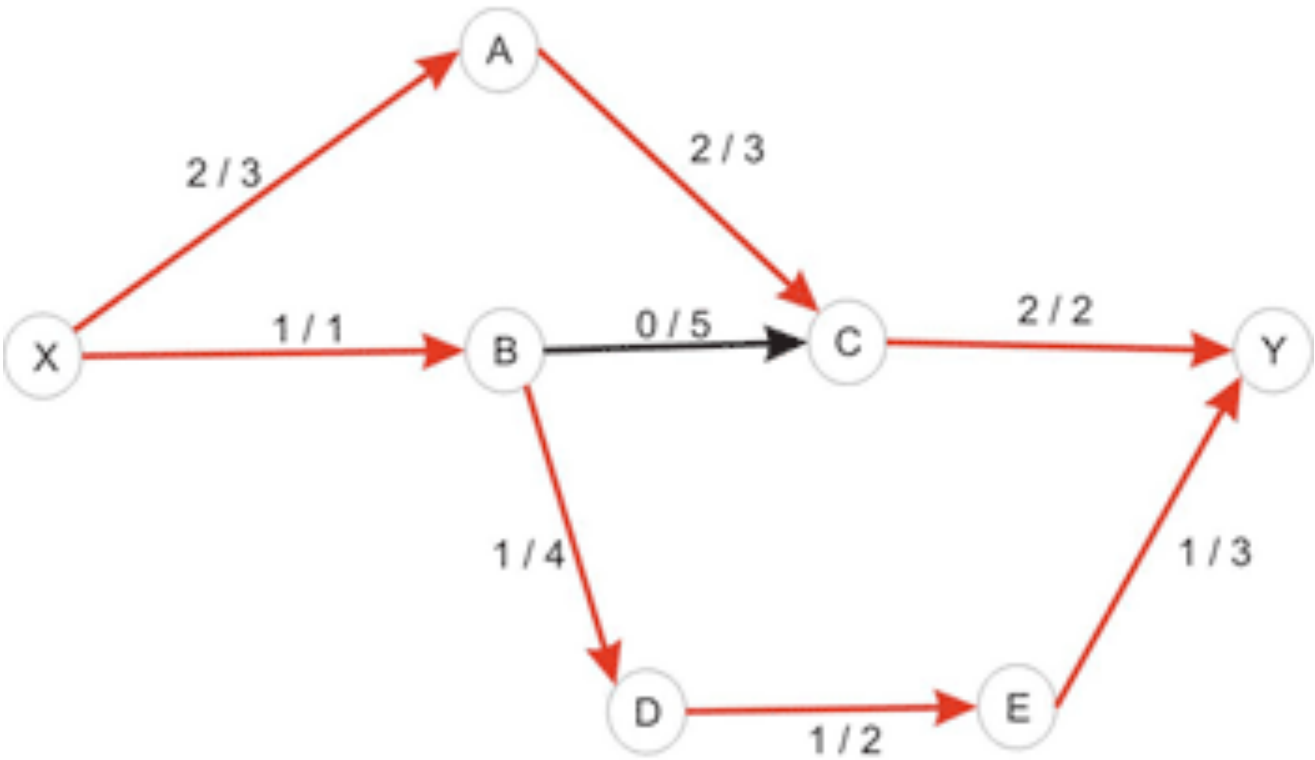


Figure 1a - Maximum Flow in a network

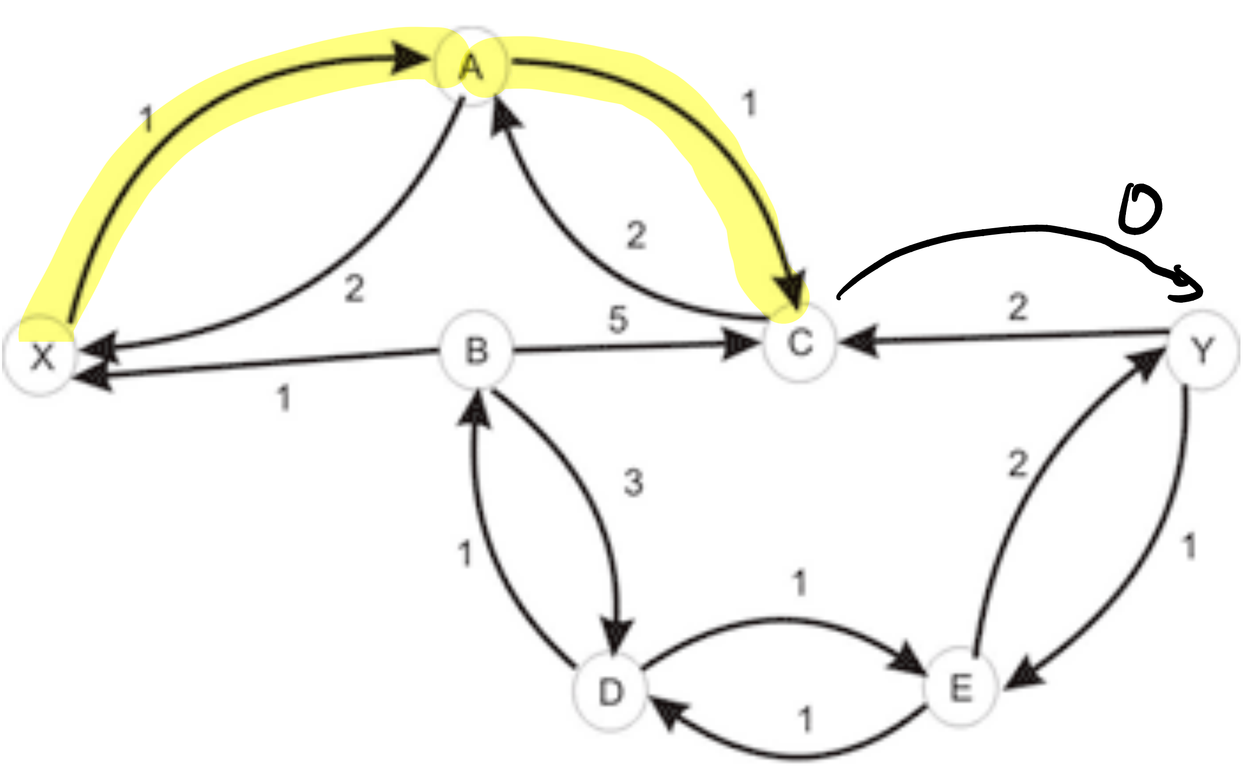
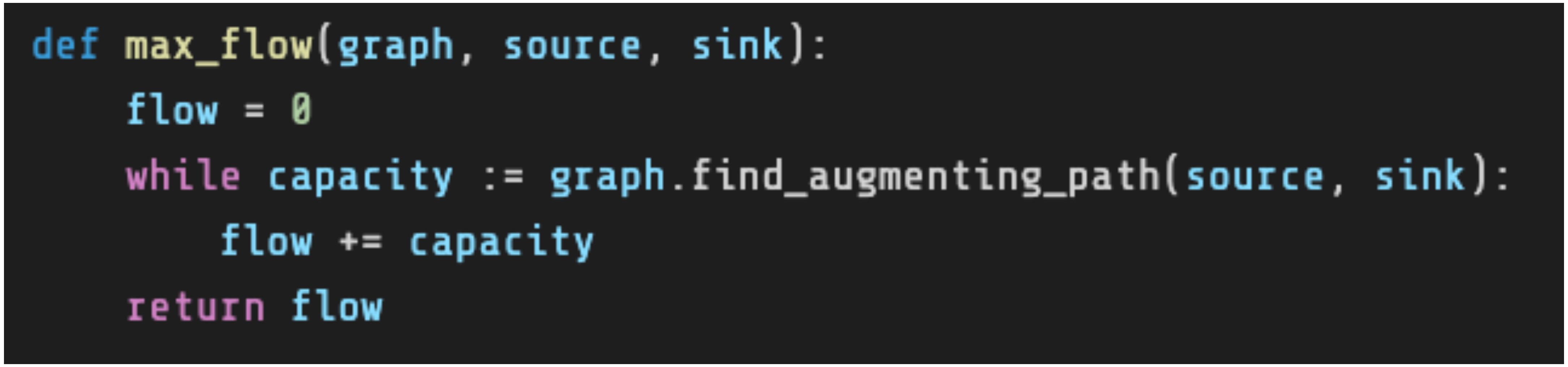


Figure 1b - The residual network of the network in 1a

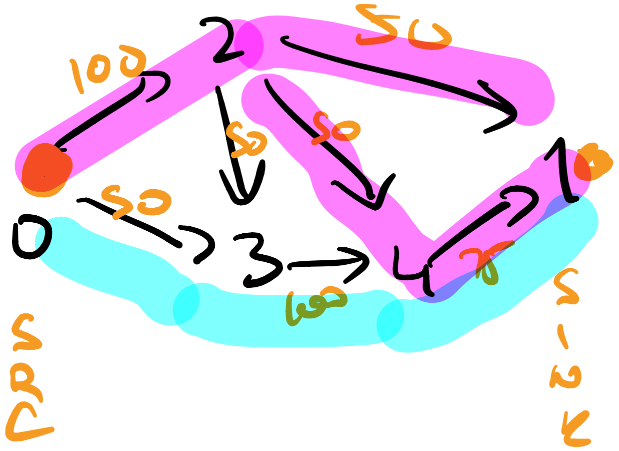
Ford-Fulkerson

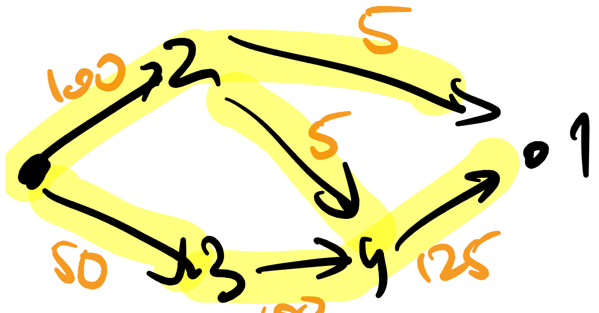


```python
def max_flow(graph, source, sink):
    flow = 0
    while capacity := graph.find_augmenting_path(source, sink):
        flow += capacity
    return flow
```

Edmonds-Karp

- Use BFS to find augmenting paths!
- $O(VE^2)$



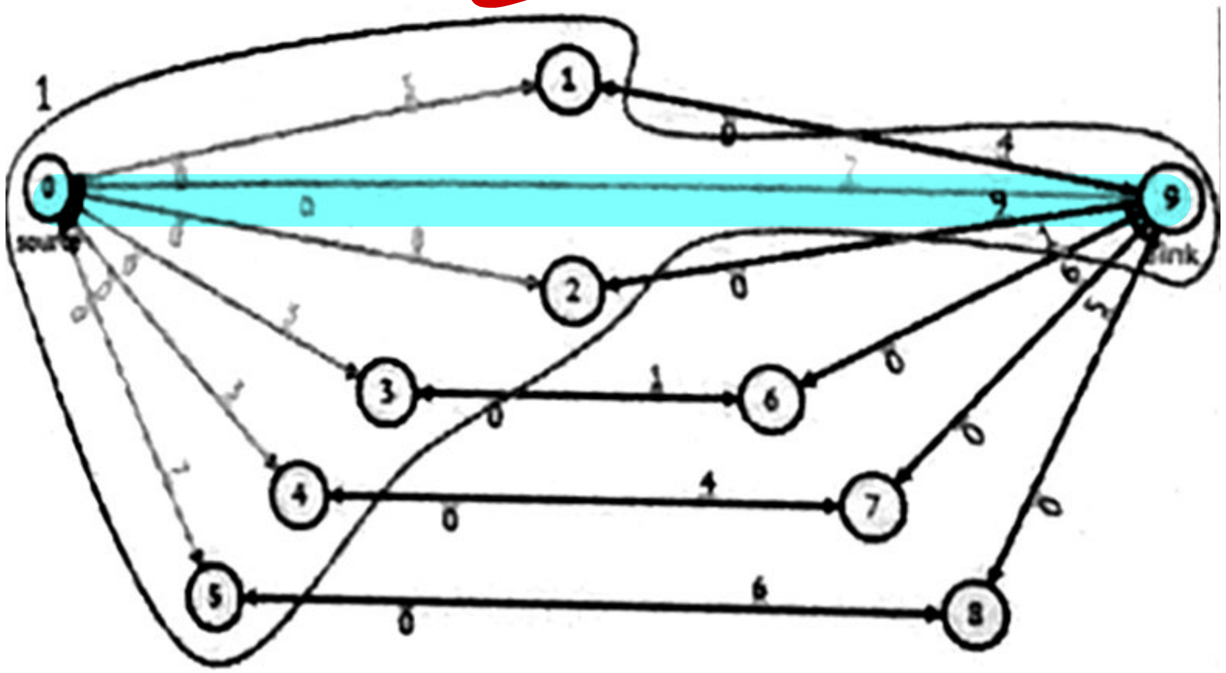


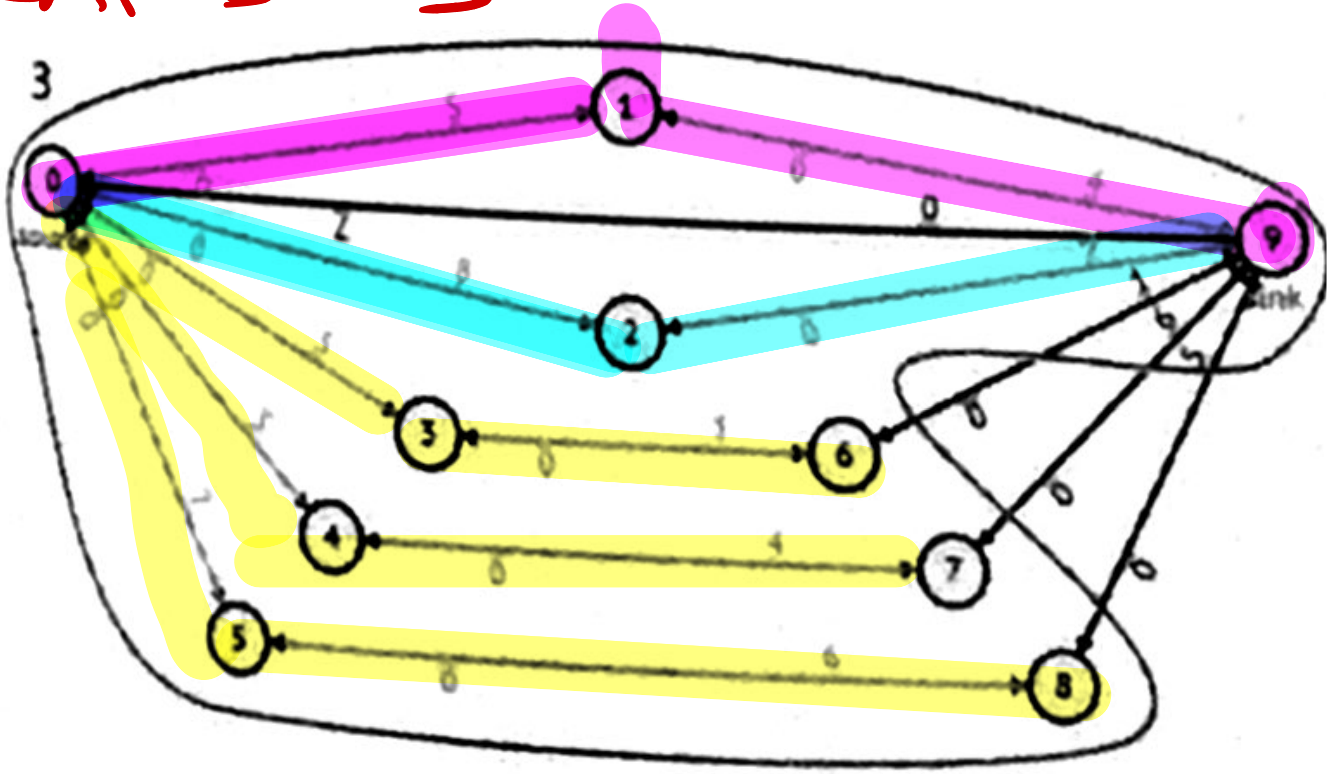
60 ✓

Dinitz's algorithm

- Use BFS to find augmenting paths!
- $O(V^2E)$ vs. $O(VE^2)_{E-K}$

LAYER GRAPHS

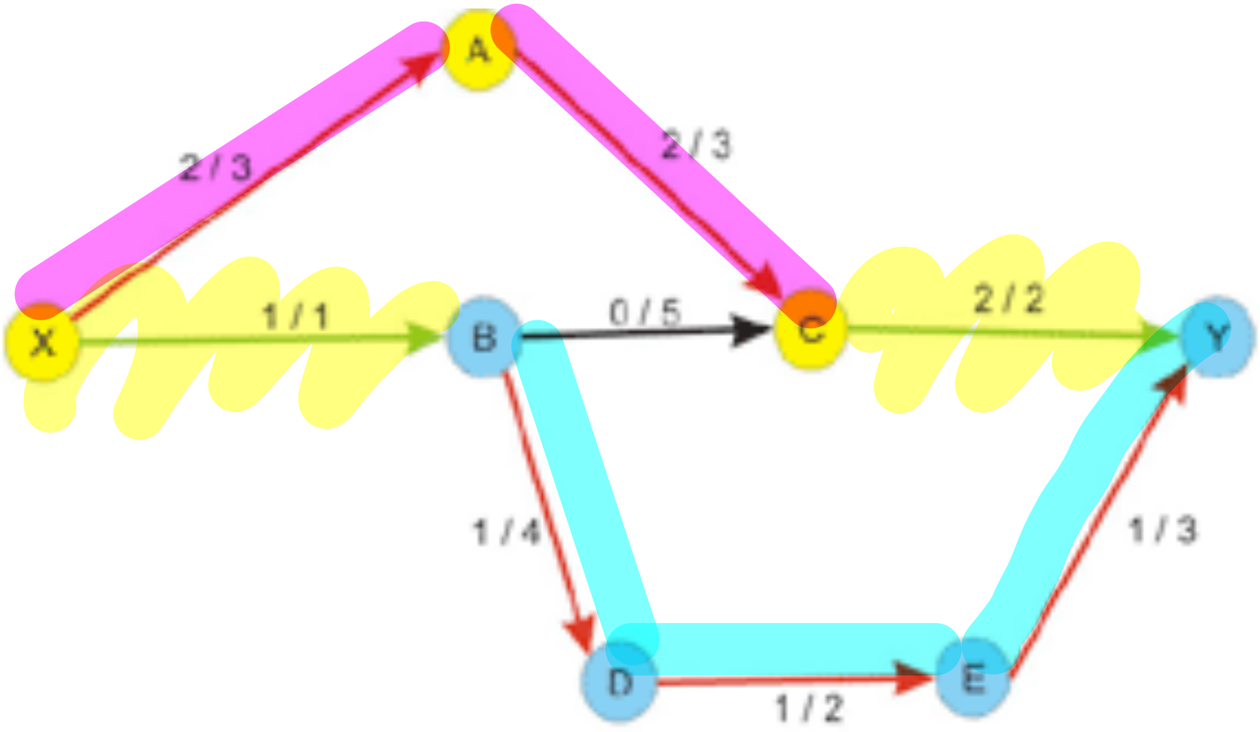




Max-flow min-cut

- Maximum flow solves another problem: min-cut in a graph (minimal set of edges that need to be removed to make a node unreachable from another node)

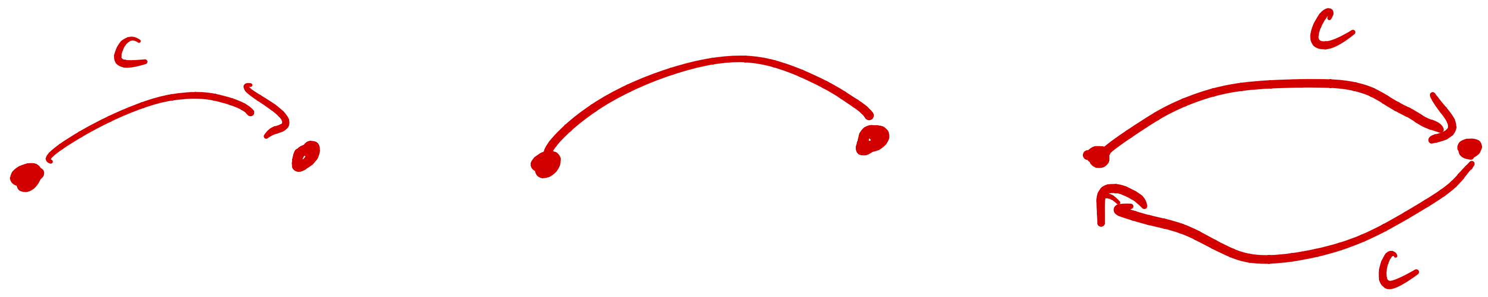
flow / capacity



The minimum cut of this network is the sum of the capacities of XB and CY and equals 3, which is also the value of the maximum flow.

Figure 5 - A minimum cut in a network

Other tricks



- Undirected graphs
- Vertex capacities

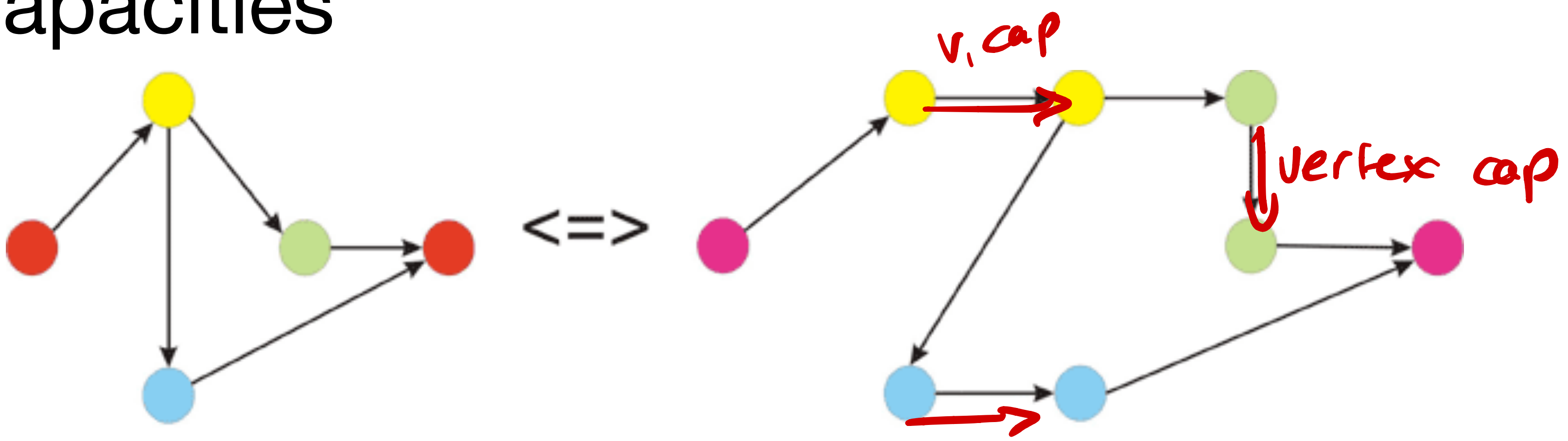


Figure 7 - Eliminating vertex-capacities

- Multi-source multi-sink

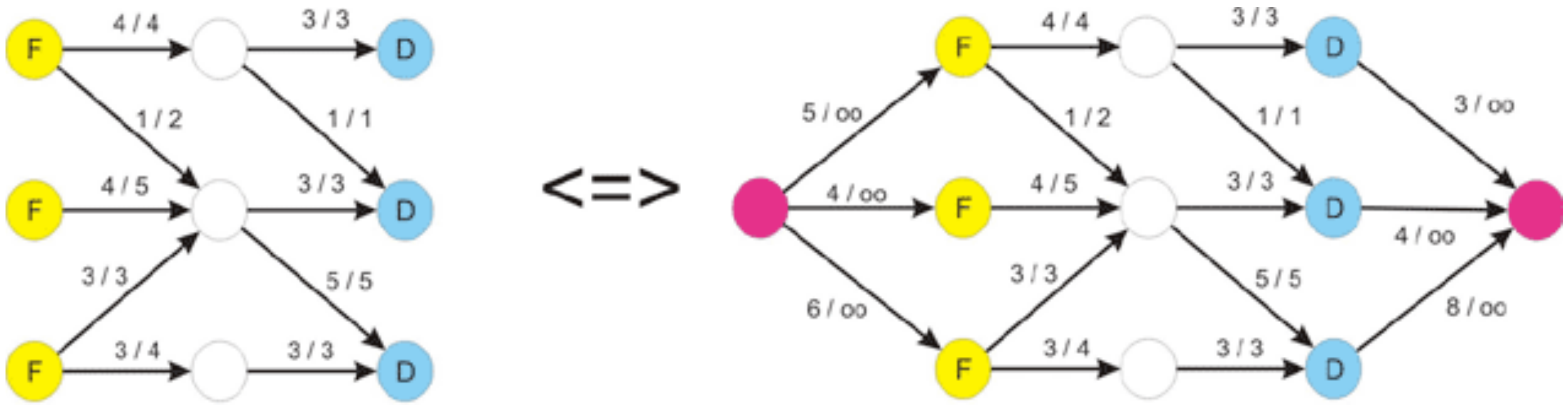
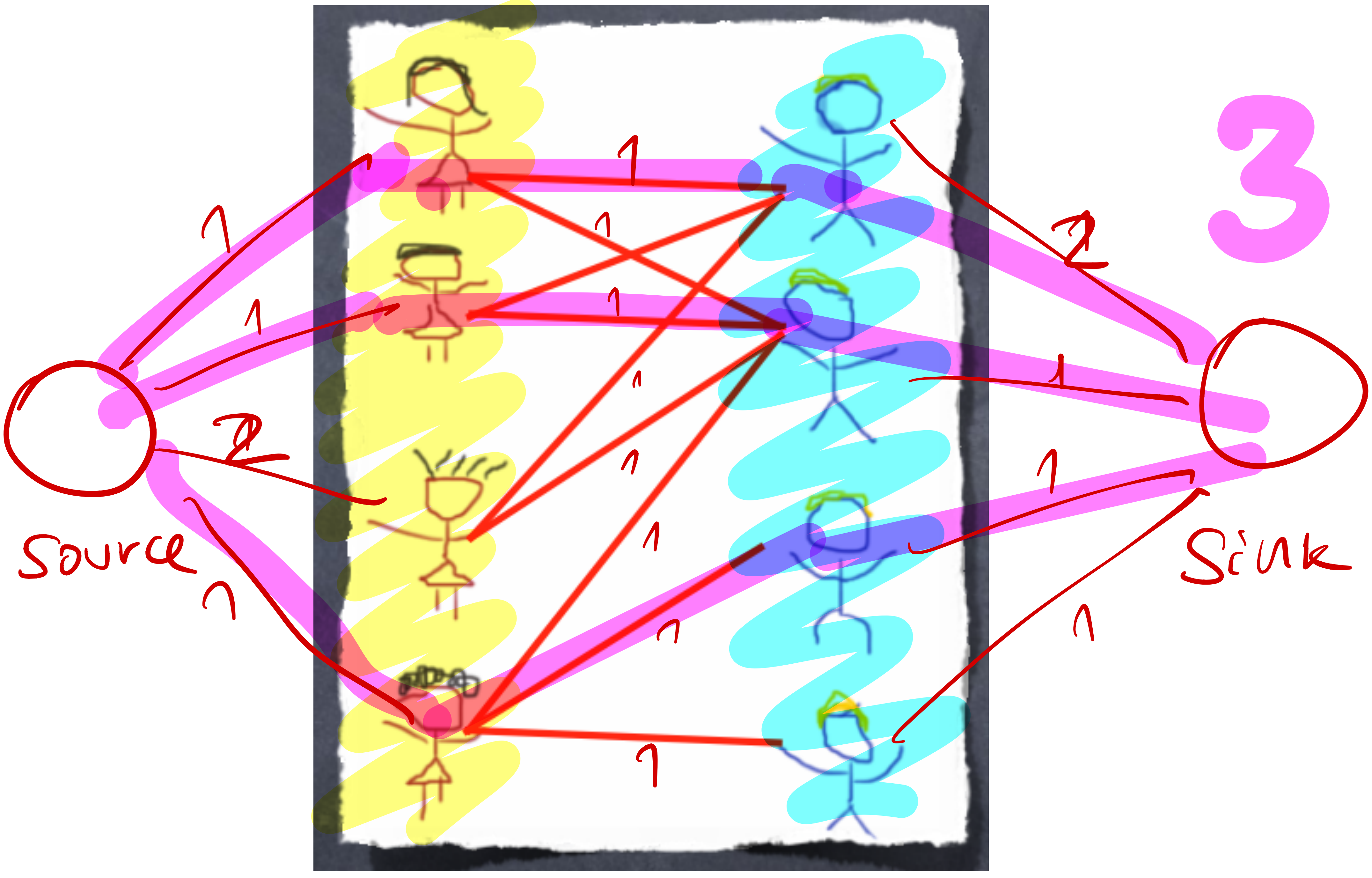


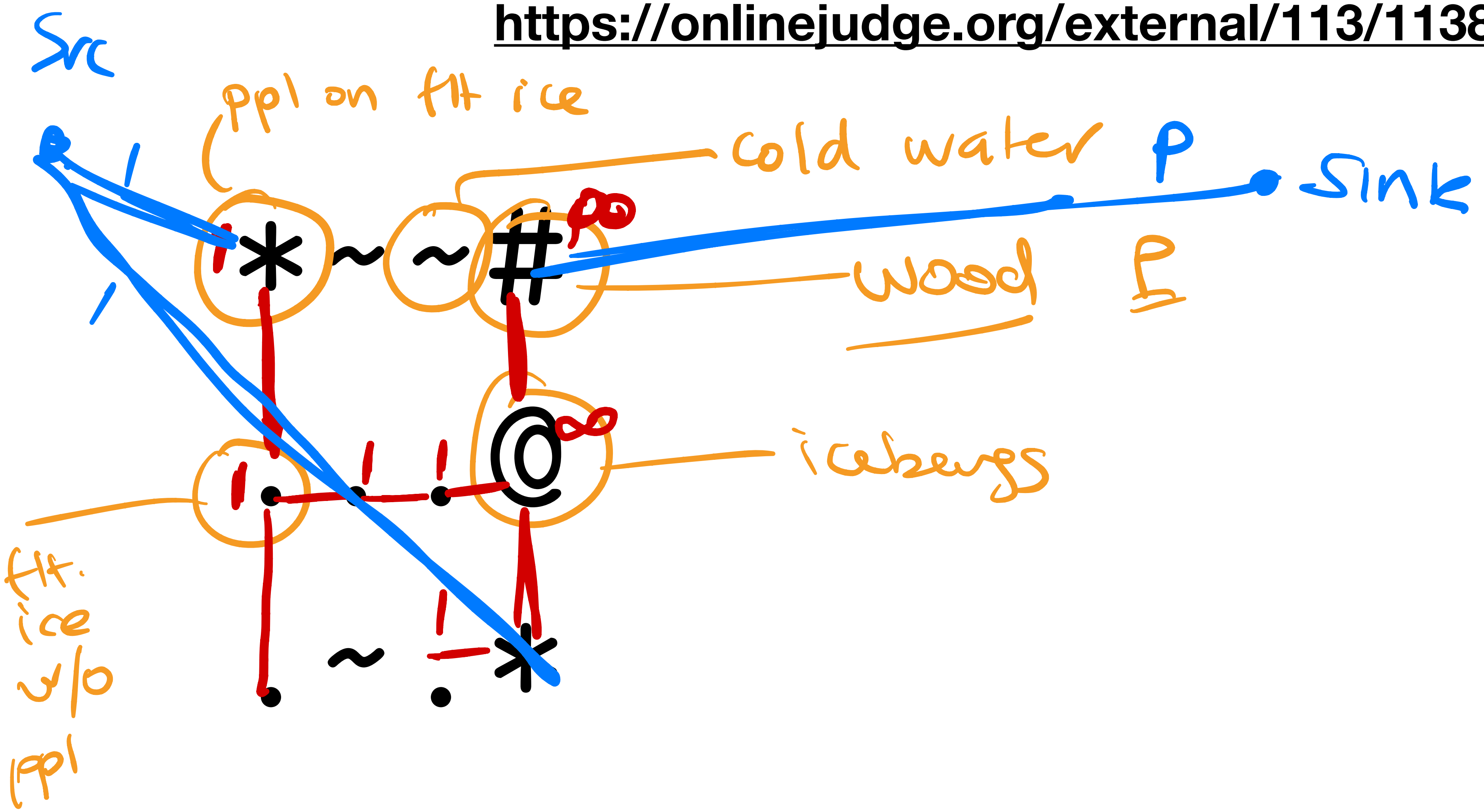
Figure 6 - Reduction of a multiple-source / multiple-sink max-flow problem

Maximum bipartite matchings



Modeling: Titanic problem

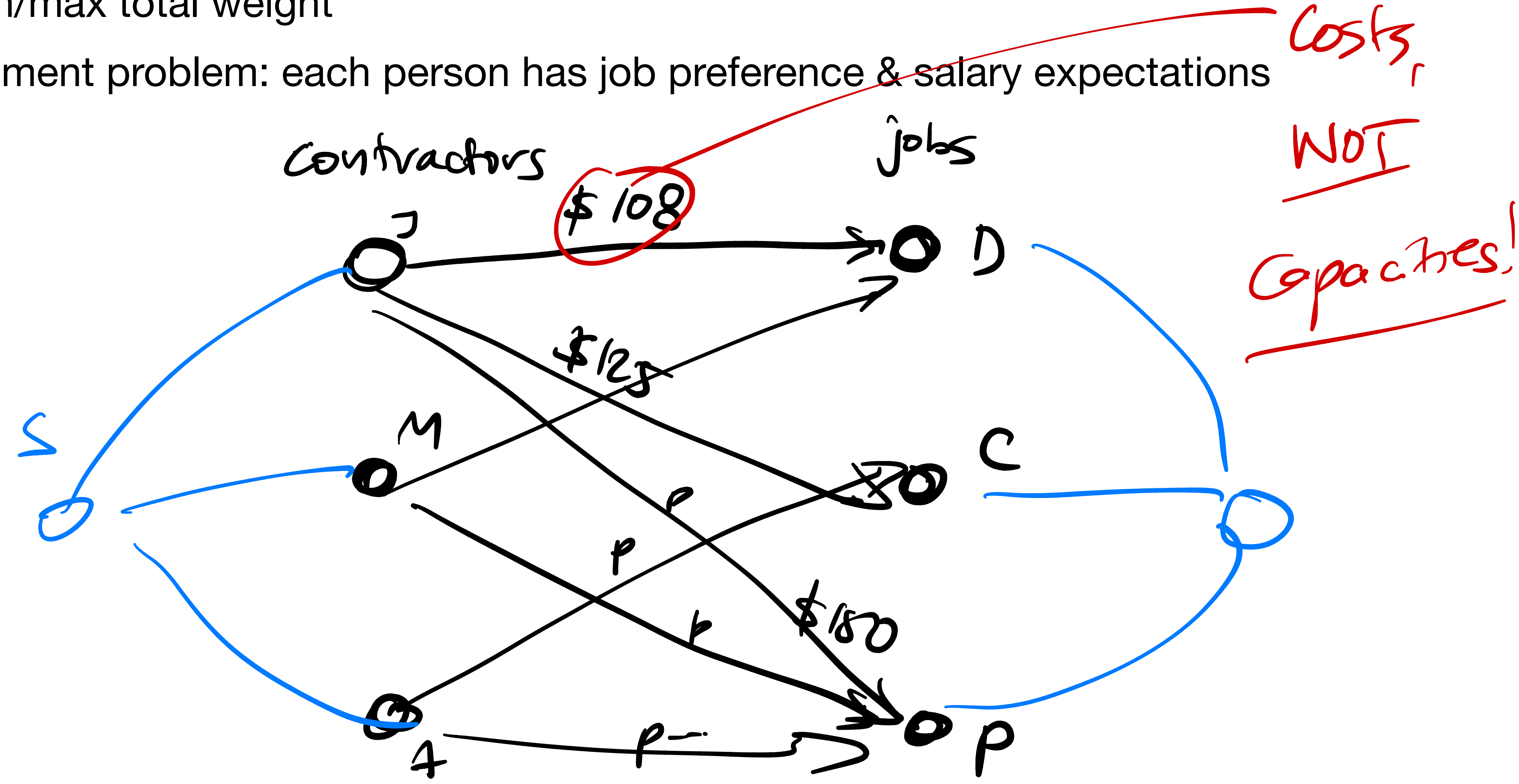
https://onlinejudge.org/external/113/11380.pdf



```
*~~#
...@
.~.*
```

Min-cost max-flow

- Sometimes edges have weights as well, and we want to pick a matching that has the min/max total weight
- Assignment problem: each person has job preference & salary expectations



Costs, NOT capacities!

$|V| \approx 450$

Problems, problems

- Network flow:
  - UVA 00259, 00820, 00563, 11167, 11380, 12873 titanic
  - Kattis: unfairplay
  - Extra:
    - CP4.2: page 434
    - UVa: 00753, 10122, 10330, 10511, 10735
    - Topcoder: rookattack, graduation, parking
- Matching:
  - UVA 11138, 00670, 12644
  - [mcmf] UVA 10594, 11301, 10746