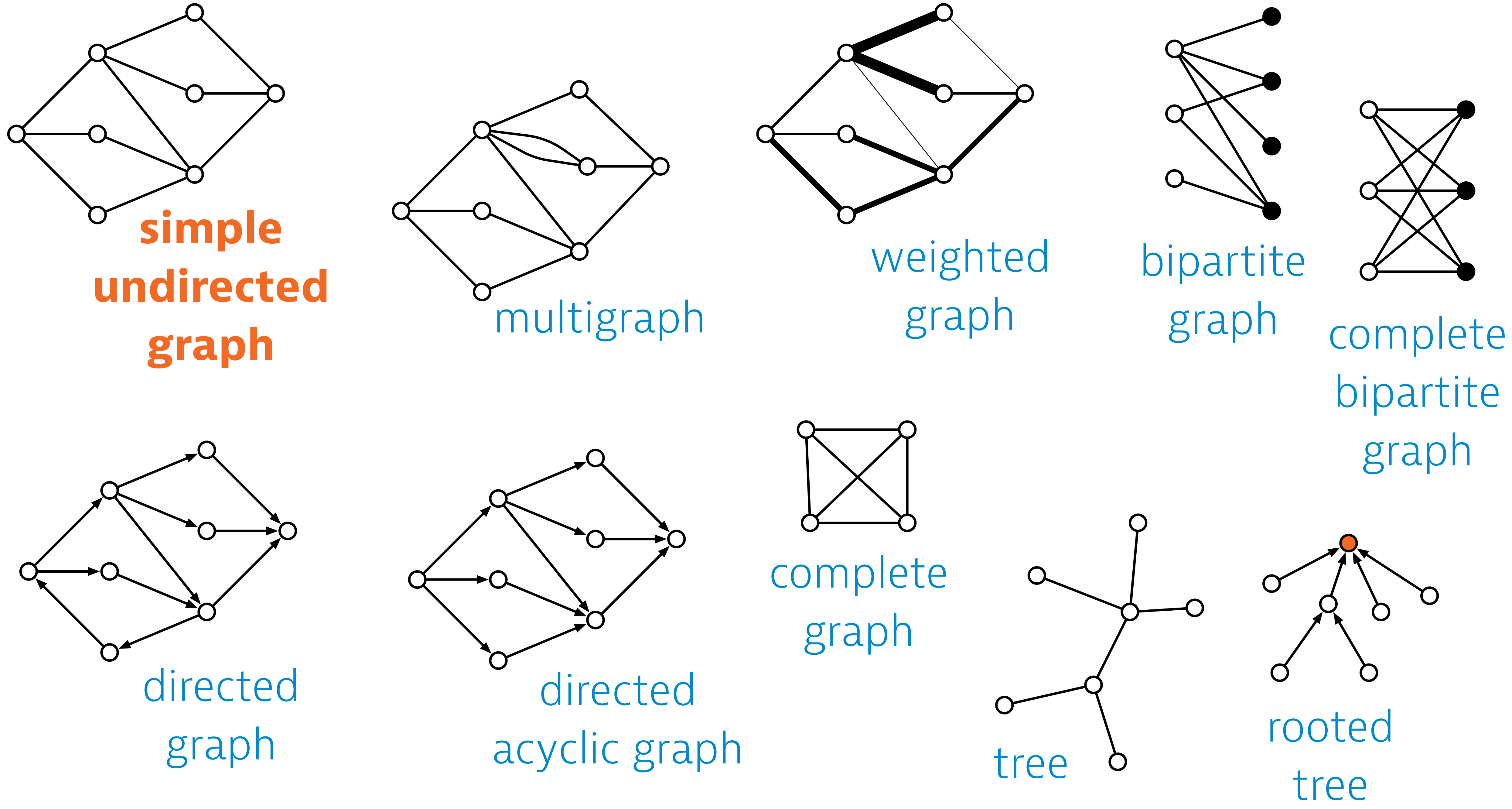
Aalto University

Competitive Programming 2020

2 Graphs, matchings, flows, cuts

Today's program

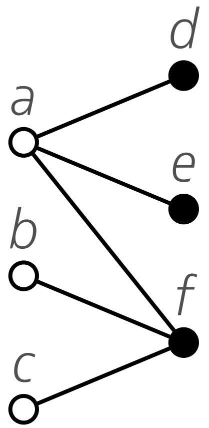
- **12:15:** Lecture (Zoom)
- **13:00:** Practice contest (CSES)
  - multiple problems to choose from
  - *try to solve at least 2 problems*
  - try to solve first on your own
  - if no progress: help available starting at **14:00**
- **16:00:** Post-contest wrap-up (Zoom)


simple undirected graph
multigraph
weighted graph
bipartite graph
complete bipartite graph
directed graph
directed acyclic graph
complete graph
tree
rooted tree

- x[i] = list of neighbors of node i
- x[i] = list of successors of node i, y[i] = list of predecessors of node i
- x[i] = list of edges incident to node i, with their weights

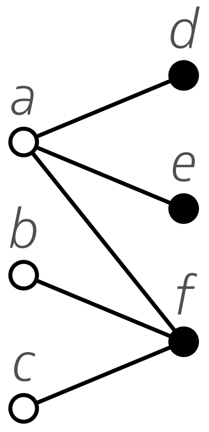
- x[i] = list of neighbors of node i
- x[i] = list of successors of node i, y[i] = list of predecessors of node i
- x[i] = list of edges incident to node i, with their weights
- x[i][j] = does there exist edge {i, j} ?
- x[i][j] = does there exist directed edge (i, j) ?
- x[i][j] = weight of the edge (i, j), 0 = no edge

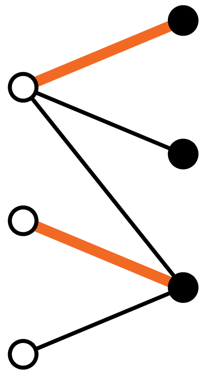
- x[i] = list of neighbors of node i
- x[i] = list of successors of node i, y[i] = list of predecessors of node i
- x[i] = list of edges incident to node i, with their weights
- x[i][j] = does there exist edge {i, j} ?
- x[i][j] = does there exist directed edge (i, j) ?
- x[i][j] = weight of the edge (i, j), 0 = no edge
- list of (i, j) pairs
- list of (i, j, w) triples
- set of (i, j) pairs
- map (i, j) → w



What is the largest **matching**?

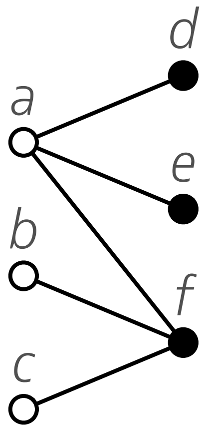
How many white–black **pairs** can you form?

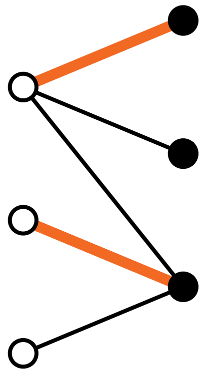




What is the largest **matching**?

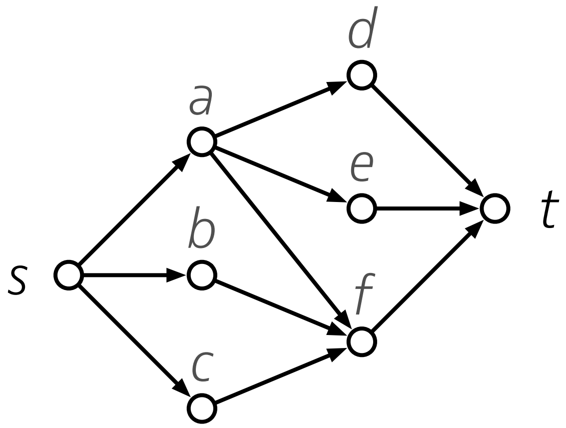
How many white–black **pairs** can you form?





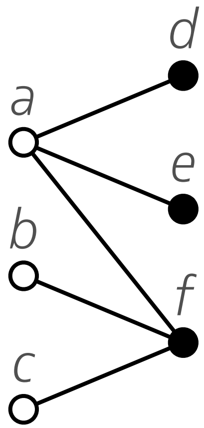
What is the largest **matching**?

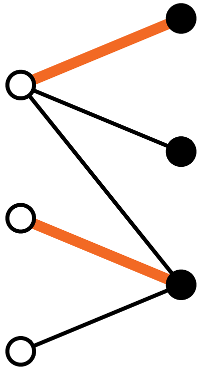
How many white–black **pairs** can you form?



How many **edge-disjoint paths** from *s* to *t* can you find?

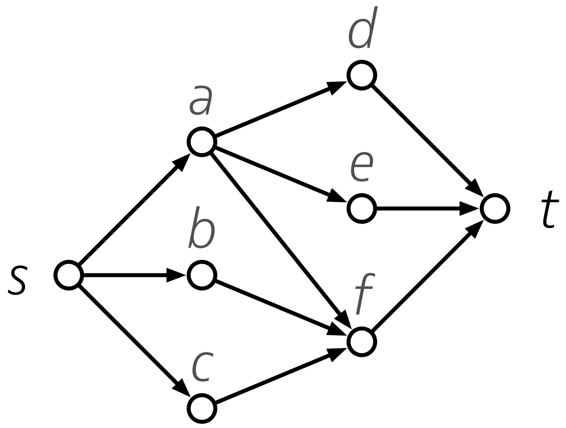
What is the largest **flow** from *s* to *t*?

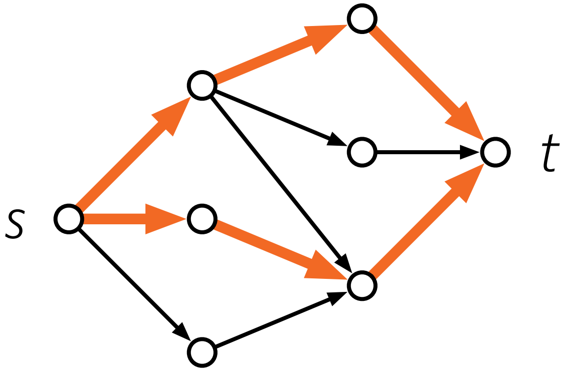




What is the largest **matching**?

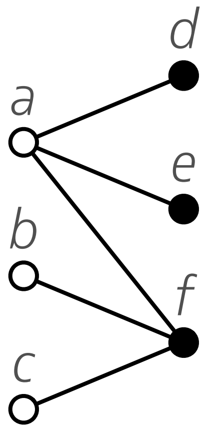
How many white–black **pairs** can you form?

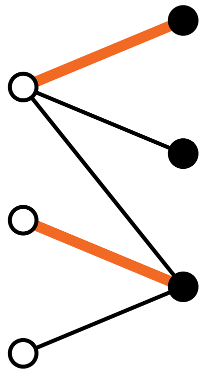




How many **edge-disjoint paths** from *s* to *t* can you find?

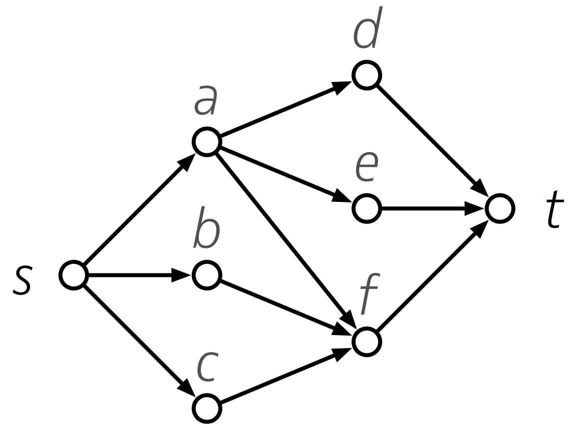
What is the largest **flow** from *s* to *t*?

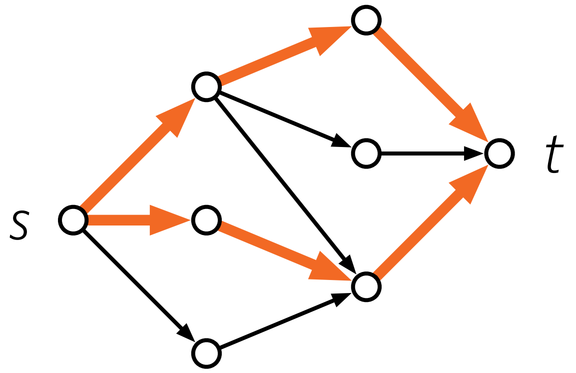




What is the largest **matching**?

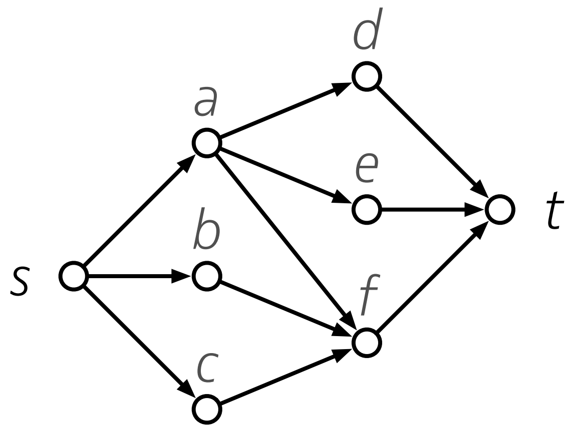
How many white–black **pairs** can you form?





How many **edge-disjoint paths** from *s* to *t* can you find?

What is the largest **flow** from *s* to *t*?

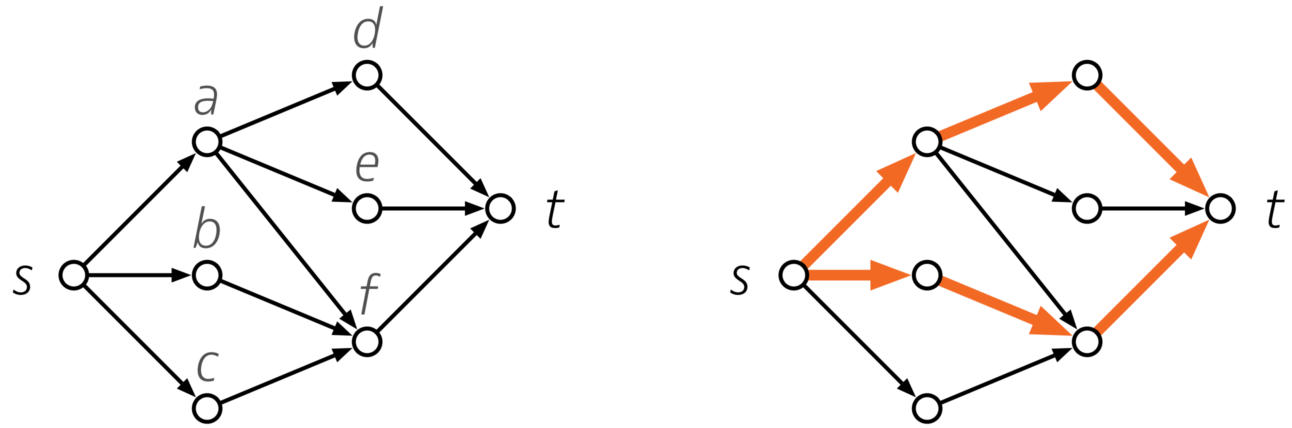


What is the smallest number of edges you need to **delete** so that there is no route from *s* to *t*?

Where is the smallest **cut**?

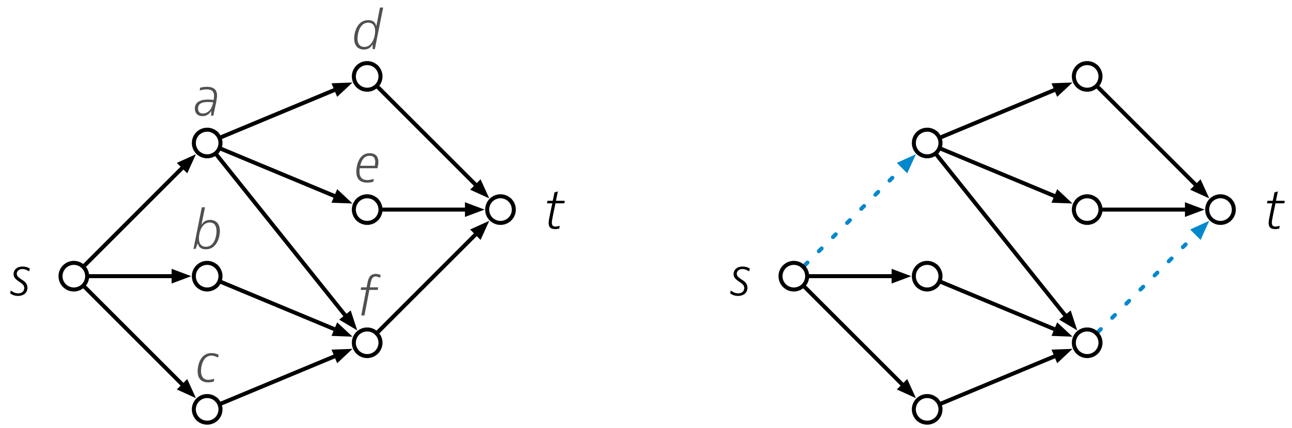
What is the largest **matching**?

How many white–black **pairs** can you form?



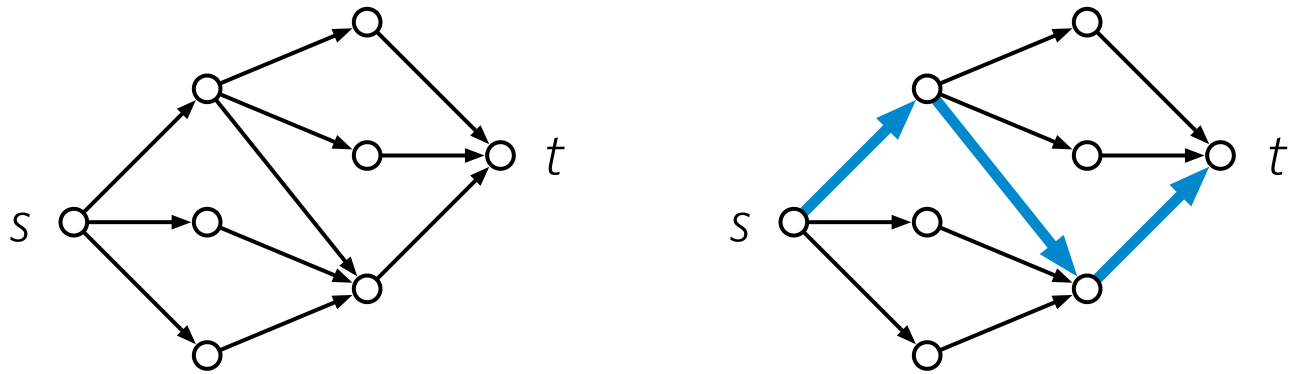
How many **edge-disjoint paths** from *s* to *t* can you find?

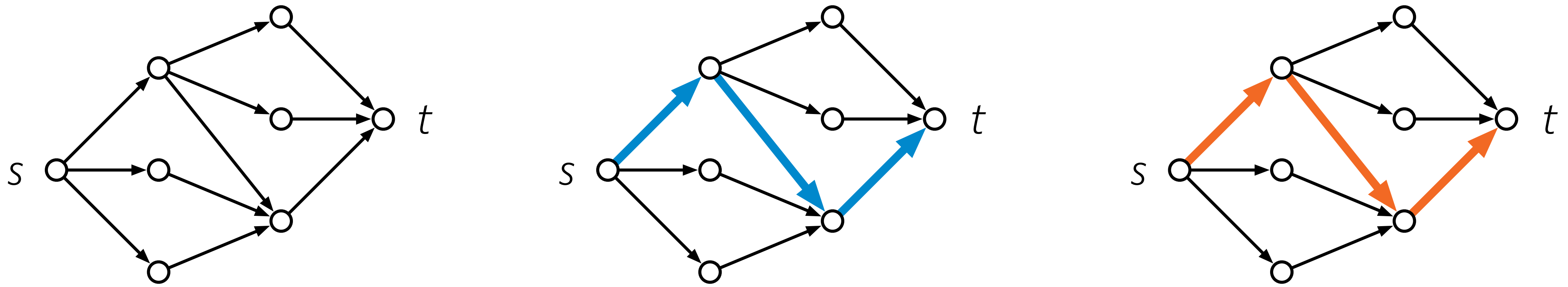
What is the largest **flow** from *s* to *t*?

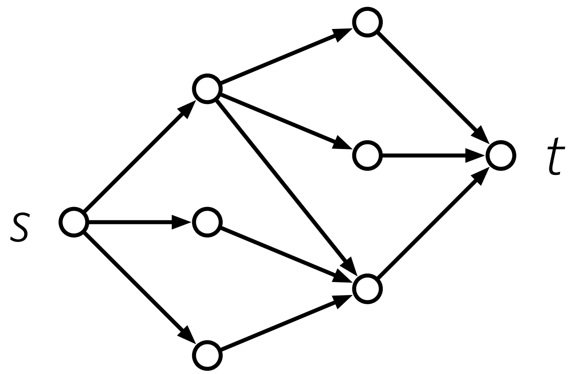


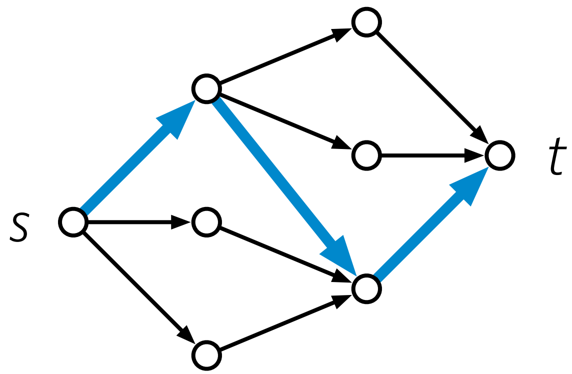
What is the smallest number of edges you need to **delete** so that there is no route from *s* to *t*?

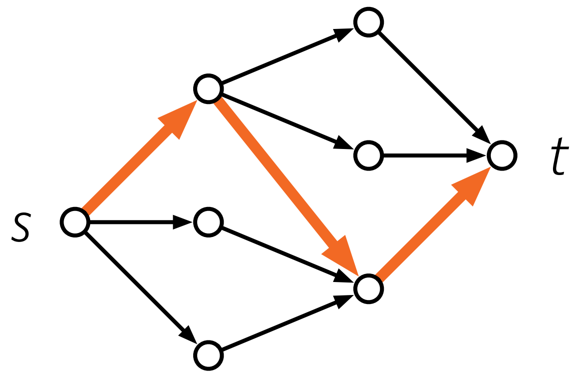
Where is the smallest **cut**?

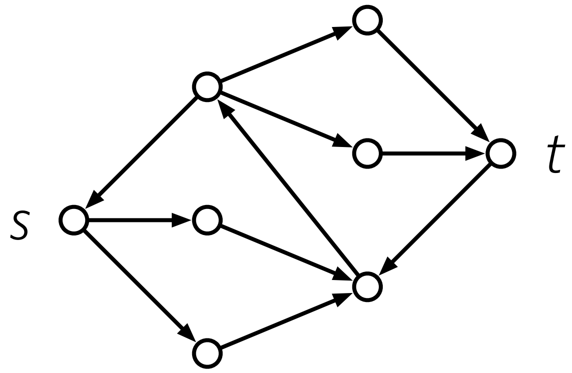

s
t
s
t

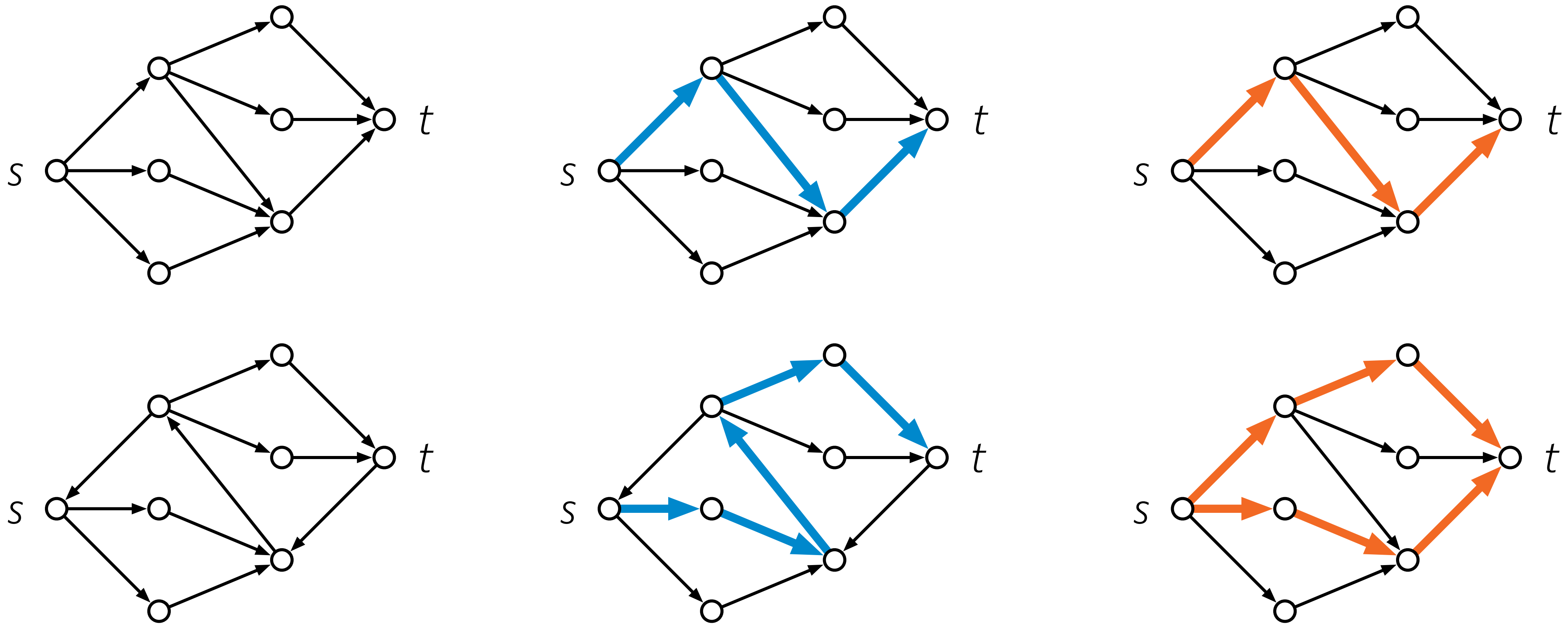

s
t
s
t
s
t

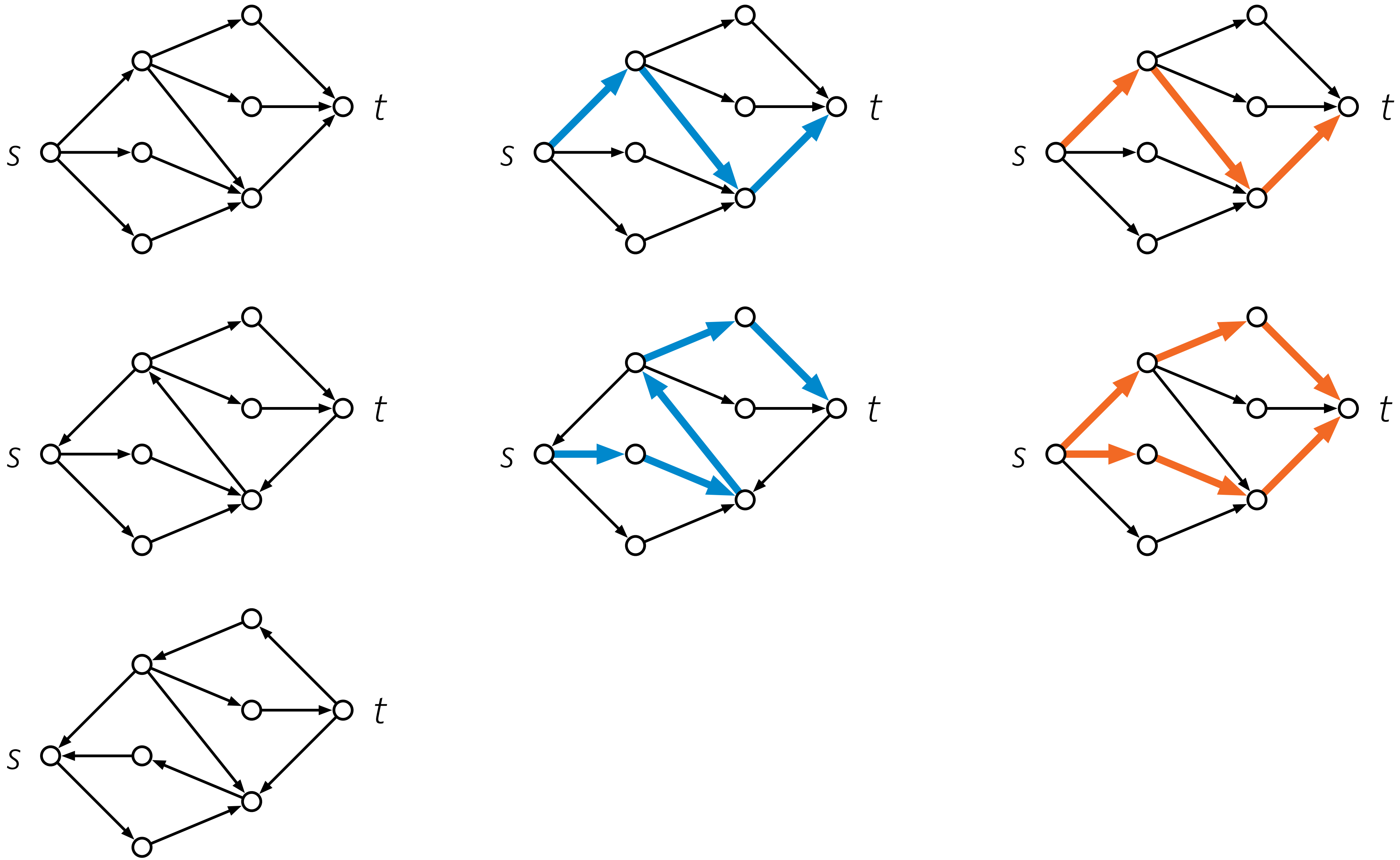

s
t

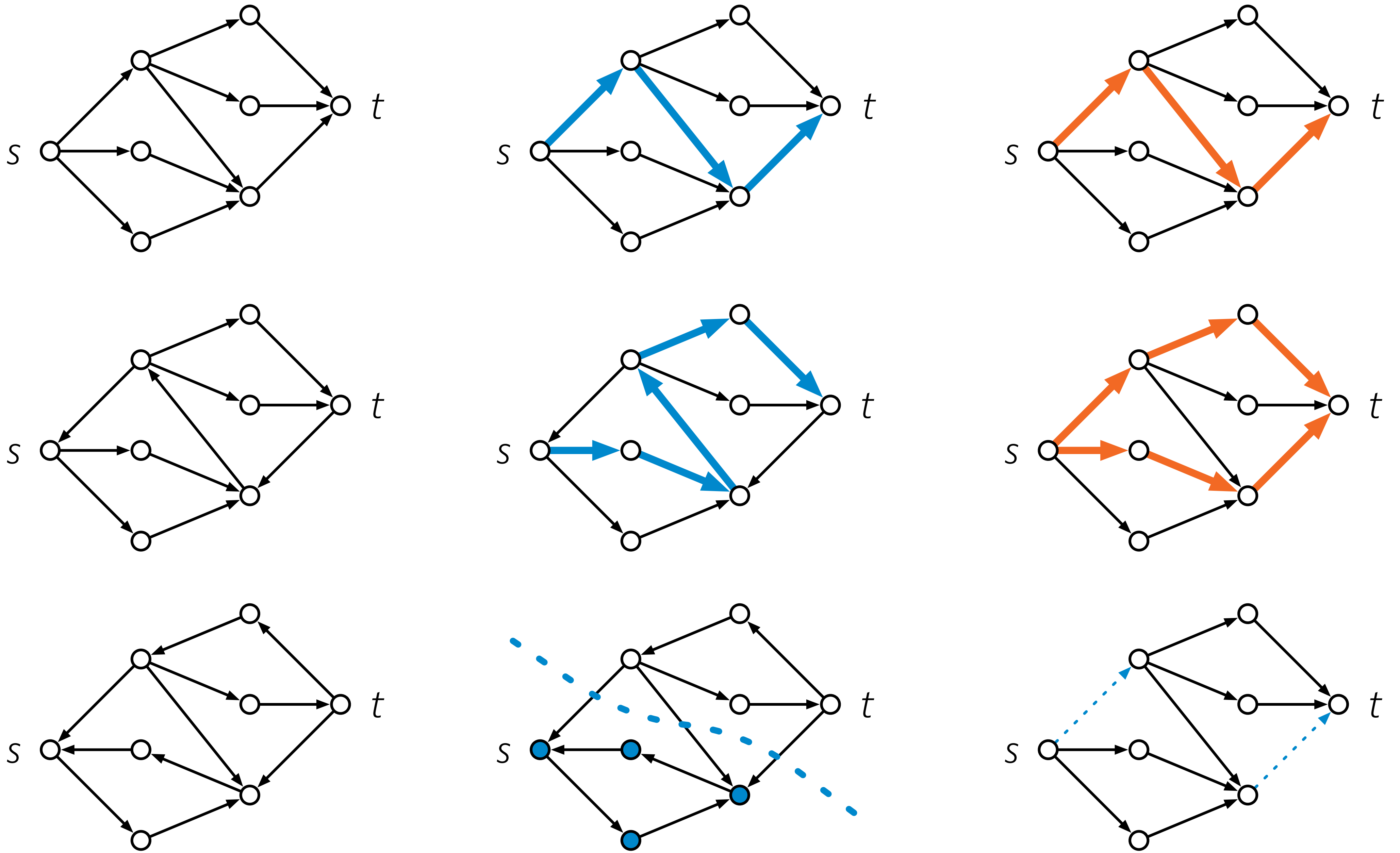

s
t

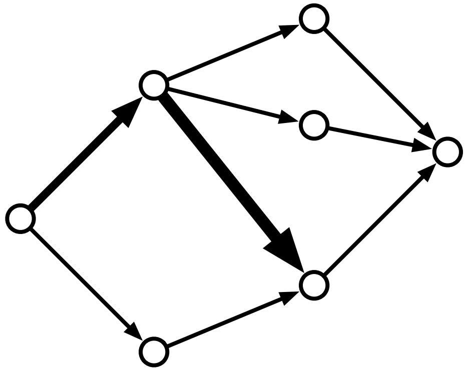

s
t


s
t


s
t
s
t
s
t
s
t
s
t
s
t

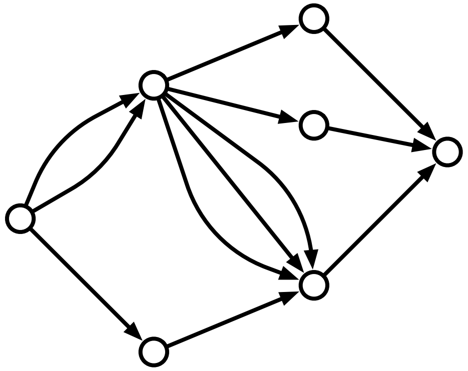

s
t


s
t
s
t
s
t
s
t
s
t
s
t
s
t
s
t
s
t



edge weights

$\approx$



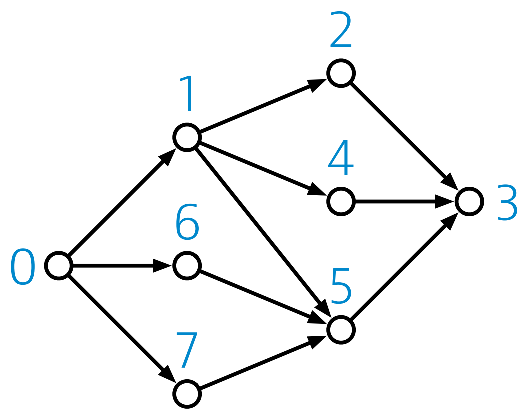
multiple parallel edges

Pick augmenting paths more carefully:

- Find **any** path (DFS) — *“Ford–Fulkerson”*
- Find **heavy** path (DFS) — *“scaling”*
- Find **short** path (BFS) — *“Edmonds–Karp”*

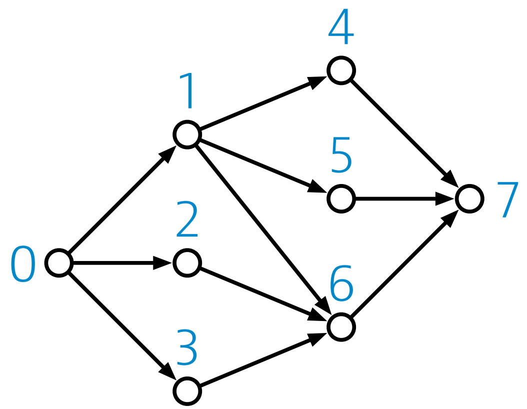
DFS = depth-first search

- when visiting a node,
  recursively visit its neighbors



BFS = breadth-first search

- when visiting a node,
  add its neighbors to a ***queue***



In both cases: **remember** what you have already visited — don’t visit them many times