Flow Conclusion

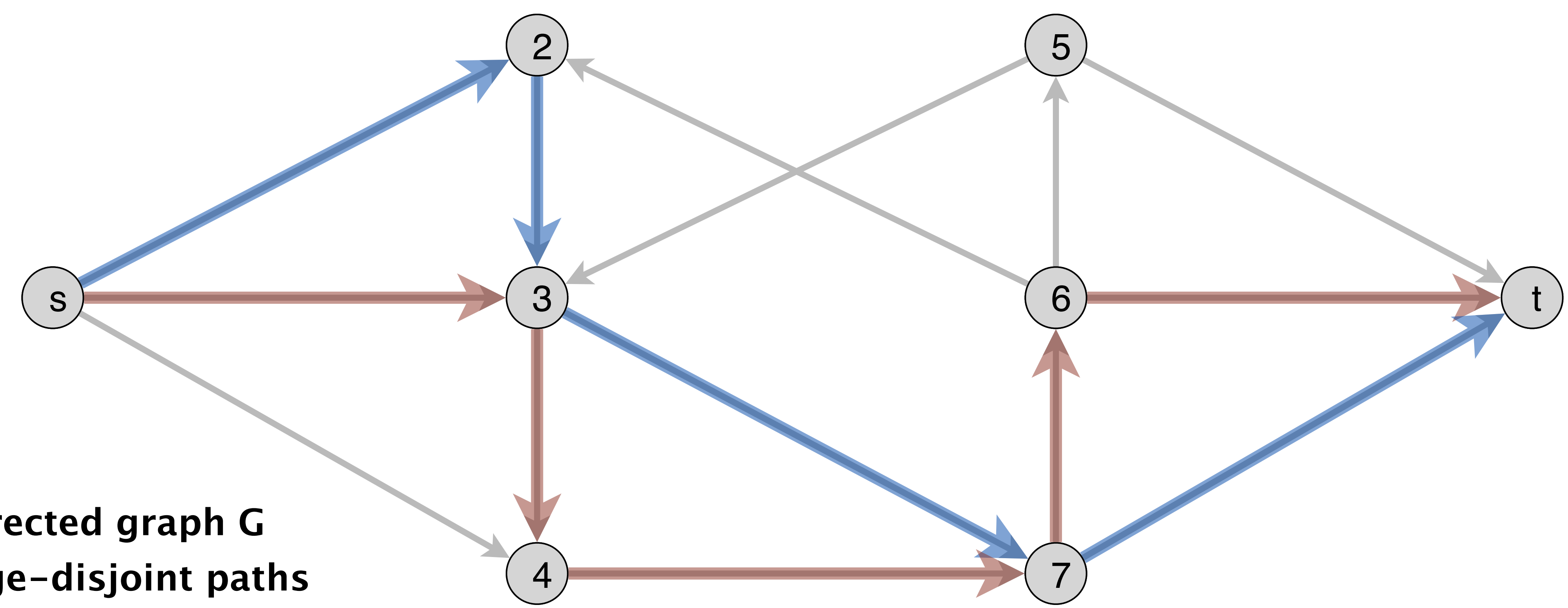
Admin

- Problem sets back (?)
  - Will probably post PS7 solutions
- Midterm on Monday!
  - Review session today
- No class next Thursday
- No daily homework until we come back

Disjoint Paths Problem

Disjoint Paths Problem

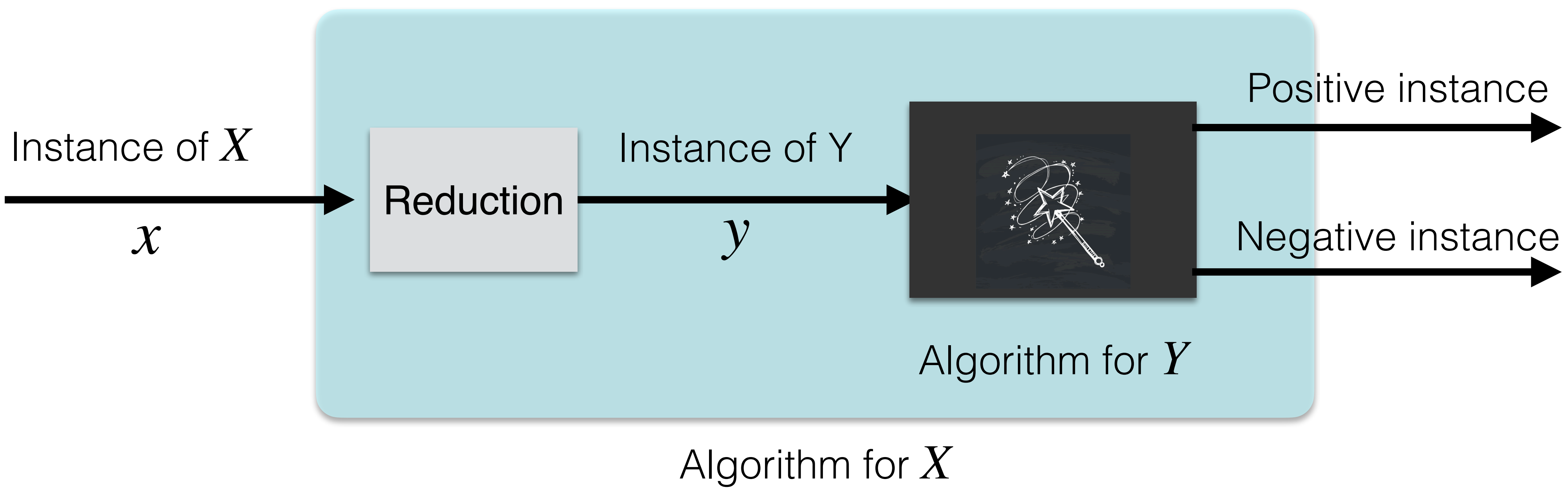
- **Definition.** Two paths are edge-disjoint if they do not have an edge in common.
- **Edge-disjoint paths problem.**
Given a directed graph with two nodes s and t, find the max number of edge-disjoint $s \rightsquigarrow t$ paths.



Directed graph G
2 edge-disjoint paths

Towards Reduction

- Given: arbitrary instance x of disjoint paths problem (X): directed graph G, with source s and sink t
- **Goal**. create a special instance y of a max-flow problem (Y): flow network $G'(V', E', c)$ with s', t' s.t.
- **1-1 correspondence.** Input graph has k edge-disjoint paths iff flow network has a flow of value k



Reduction to Max Flow

- **Reduction.** G' : same as G with unit capacity assigned to every edge
- **Claim** [Correctness of reduction]. G has k edge disjoint $s \rightsquigarrow t$ paths iff G' has an integral flow of value k.
- Proof. $(\Rightarrow)$
- Set $f(e) = 1$ if e in some disjoint $s \rightsquigarrow t$, $f(e) = 0$ otherwise.
- Consider cut $(\{s\}, V - \{s\})$, we get that $v(f) = k$
- Why is f feasible? Capacity constraint? Conservation?
  - We only ever send 1 unit of flow, so capacity is never violated
  - Say node u is part of $k' \leq k$ paths, then $f_{in}(u) = f_{out} = k'$

Reduction to Max Flow

- **Reduction.** G' : same as G with unit capacity assigned to every edge
- **Claim** [Correctness of reduction]. G has k edge disjoint $s \rightsquigarrow t$ paths iff G' has an integral flow of value k.
- Proof. $(\Rightarrow)$
- Set $f(e) = 1$ if e in some disjoint $s \rightsquigarrow t$, $f(e) = 0$ otherwise.
- Consider cut $(\{s\}, V - \{s\})$, we get that $v(f) = k$
- We argued that f is a feasible flow with $v(f) = k$ ∎
- $(\Leftarrow)$ Need to show: If G' has a flow of value k then there are k edge-disjoint $s \rightsquigarrow t$ paths in G

Correction of Reduction

- **Claim**. $(\Leftarrow)$ If f is a 0-1 flow of value k in G', then the set of edges where $f(e) = 1$ contains a set of k edge-disjoint $s \rightsquigarrow t$ paths in G.
- **Proof** [By induction on the # of edges k' with $f(e) = 1$]
- If $k' = 0$, no edges carry flow, nothing to prove
- IH: Assume claim holds for all flows that use $< k'$ edges
- Consider an edge $s \to u$ with $f(s \to u) = 1$
- By flow conservation, there exists an edge $u \to v$ with $f(u \to v) = 1$, continue "tracing out the path" until
- Case (a) reach t, Case (b) visit a vertex v for a 2nd time

Correction of Reduction

- **Case (a)** We reach t, then we found a $s \rightsquigarrow t$ path P
  - f' : Decrease the flow on edges of P by 1
  - $v(f') = v(f) - 1 = k - 1$
  - Number of edges that carry flow now $< k'$: can apply IH and find $k - 1$ other $s \rightsquigarrow t$ disjoint paths
- **Case (b)** visit a vertex v for a 2nd time: consider cycle C of edges visited btw 1st and 2nd visit to v
  - f' : decrease flow values on edges in C to zero
  - $v(f') = v(f)$ but # of edges in f' that carry flow $< k'$, can now apply IH to get k edge disjoint paths

$\blacksquare$

Summary & Running Time

- Proved k edge-disjoint paths iff flow of value k
- Thus, max-flow iff max # of edge-disjoint $s \rightsquigarrow t$ paths
- Running time of algorithm overall:
    - Running time of reduction + running time of solving the max-flow problem (dominates)
- What is running time of Ford–Fulkerson algorithm for a flow network with all unit capacities?
    - $O(nm)$
- Overall running time of finding max # of edge-disjoint $s \rightsquigarrow t$ paths: $O(nm)$

Midterm 2

- No network flow **reductions** on the midterm
- There may be a question about what a flow is/Ford Fulkerson/max flow-min cut
- (See the practice midterm)