Discrete Optimization
Lecture 2

Part 2

st Maxflow

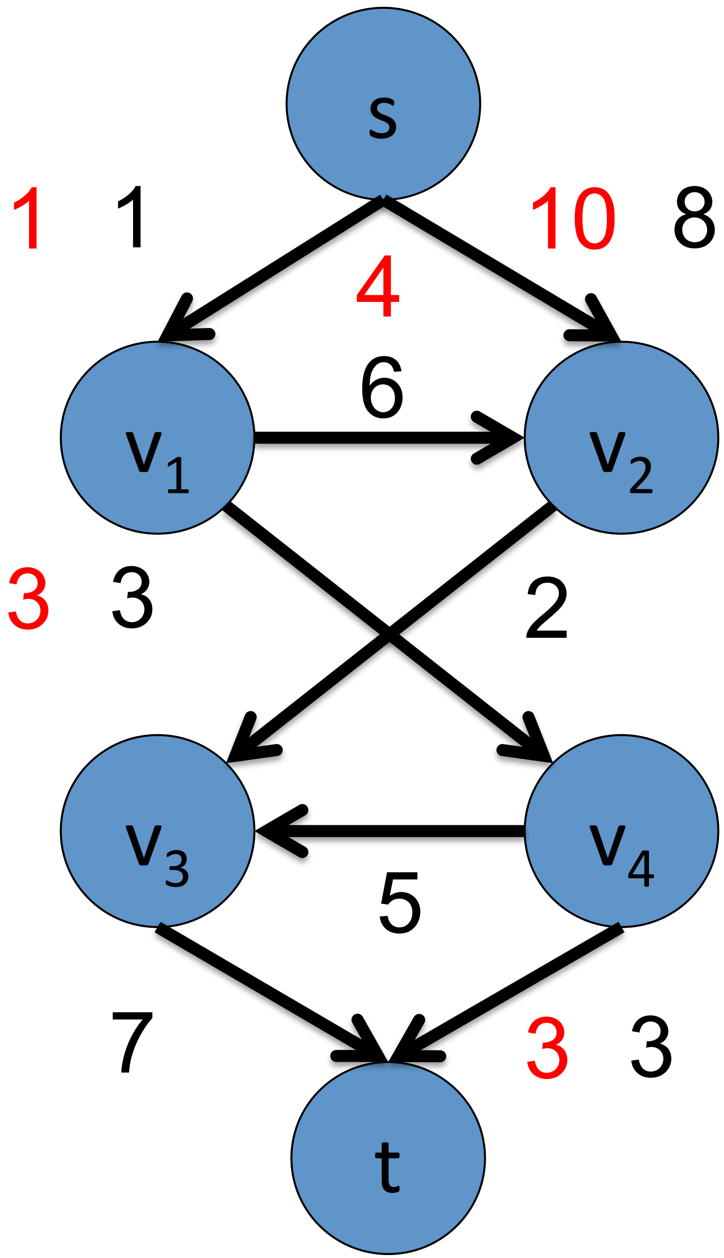
Slides online: https://project.inria.fr/2015ma2827

Slides courtesy of M. Pawan Kumar

Outline

- Preliminaries
  - Functions and Excess Functions
  - s-t Flow
  - s-t Cut
  - Flows vs. Cuts
- Maximum Flow
- Algorithms

Functions on Arcs



D = (V, A)

Arc capacities c(a)

Function f: A ➔ Reals

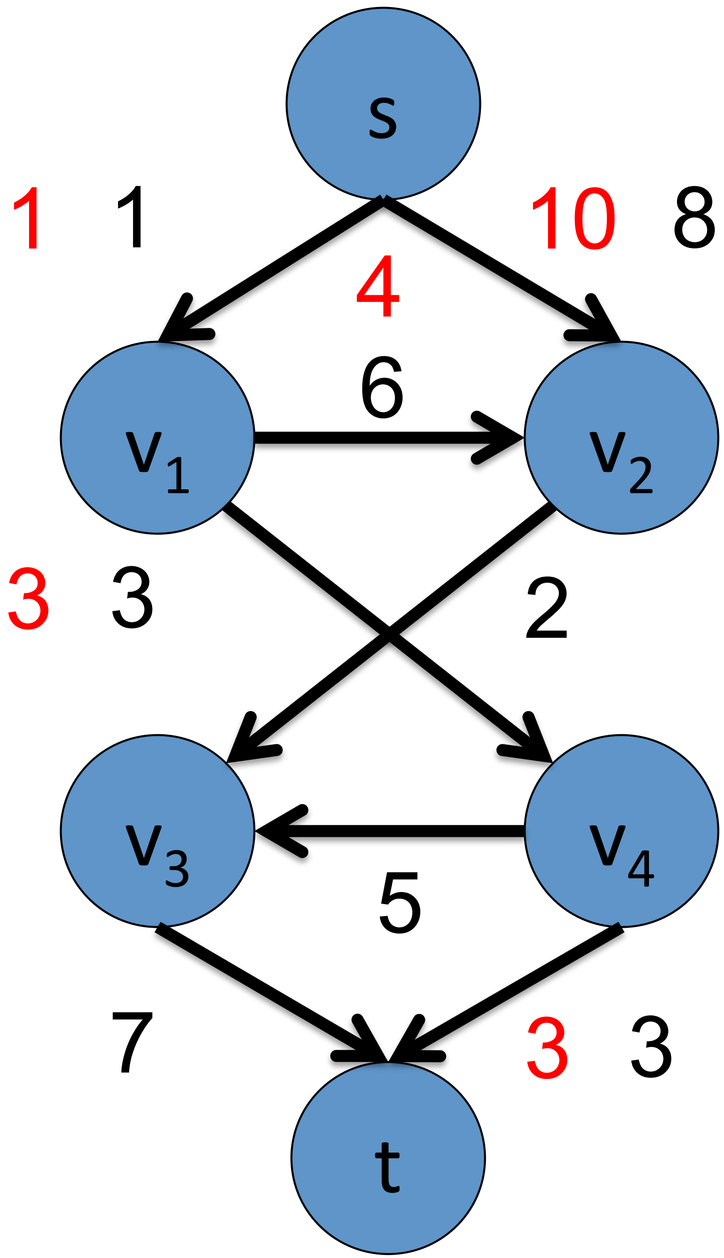
Excess function $E_f(v)$

Incoming value

\-

Outgoing value

Functions on Arcs



D = (V, A)

Arc capacities c(a)

Function f: A ➔ Reals

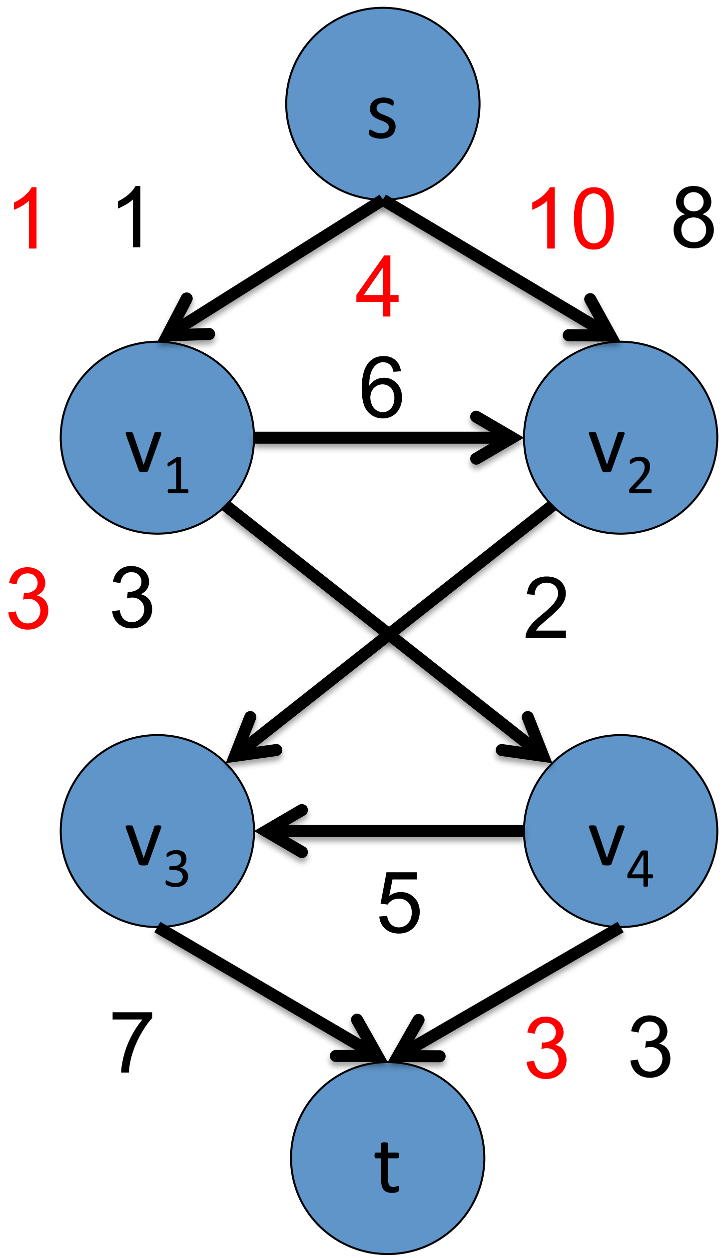
Excess function $E_f(v)$

$\Sigma_{a \in \text{in-arcs}(v)} f(a)$

-

Outgoing value

Functions on Arcs



D = (V, A)

Arc capacities c(a)

Function f: A ➔ Reals

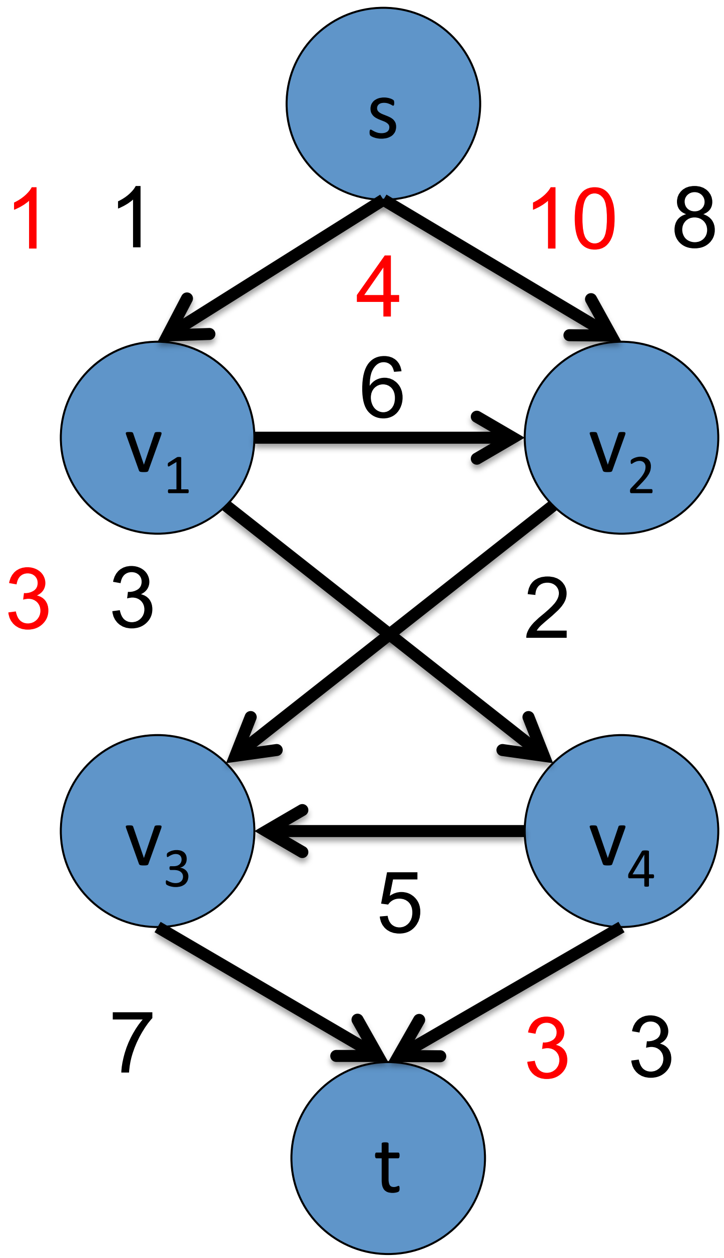
Excess function $E_f(v)$

$$\Sigma_{a \in \text{in-arcs}(v)} f(a)$$

-

$$\Sigma_{a \in \text{out-arcs}(v)} f(a)$$

Functions on Arcs



D = (V, A)

Arc capacities c(a)

Function f: A → Reals

Excess function $E_f(v)$

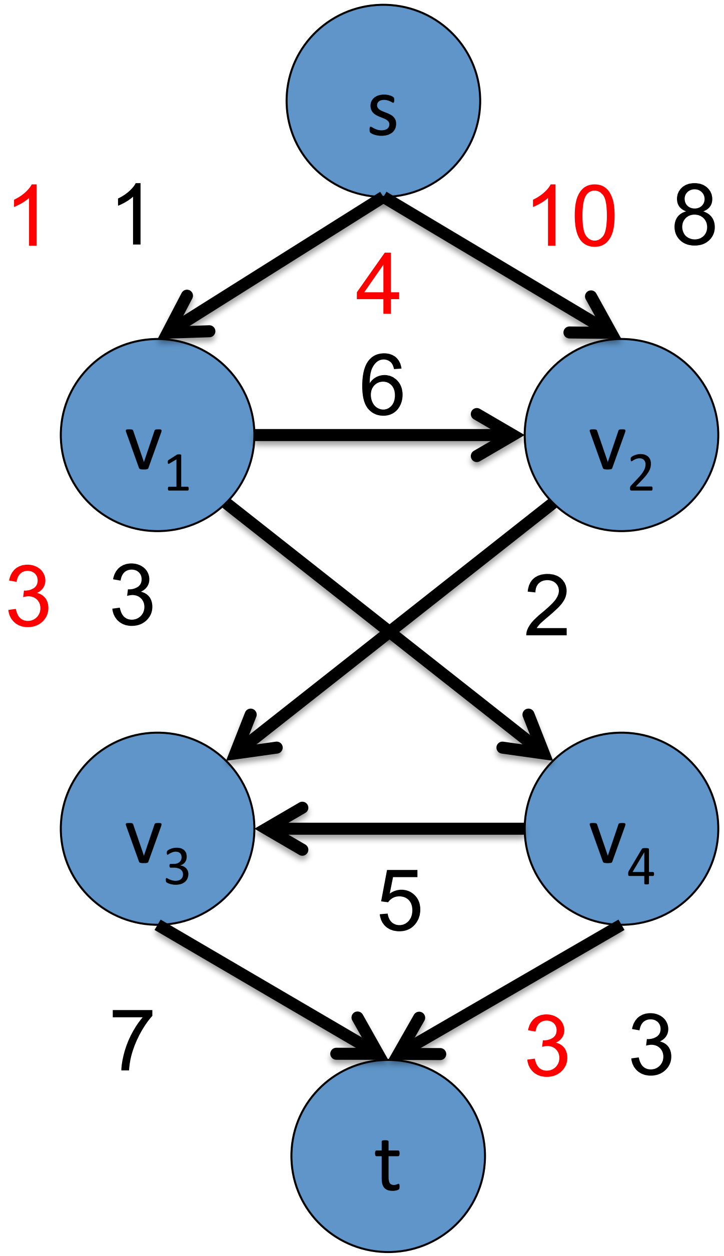
f(in-arcs(v))

\-

f(out-arcs(v))

$E_f(v_1)$ -6

Functions on Arcs



D = (V, A)

Arc capacities c(a)

Function f: A ➔ Reals

Excess function $E_f(v)$

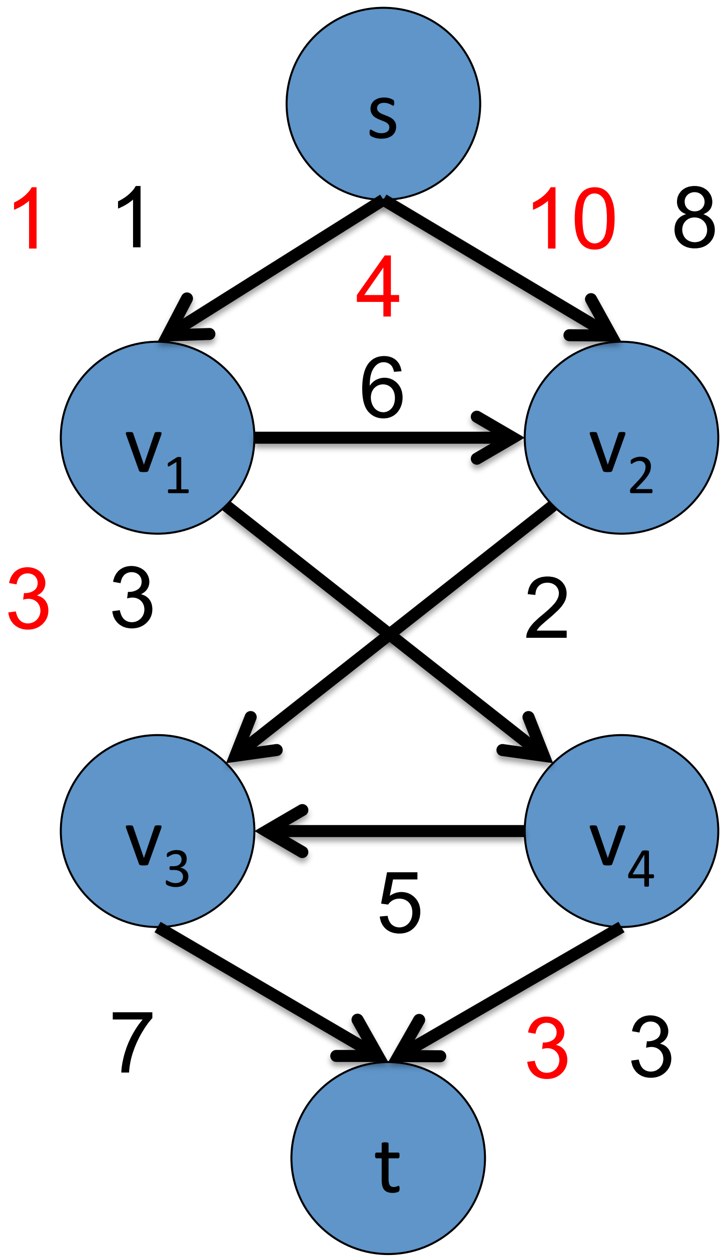
f(in-arcs(v))

-

f(out-arcs(v))

$E_f(v_2)$ 14

Excess Functions of Vertex Subsets



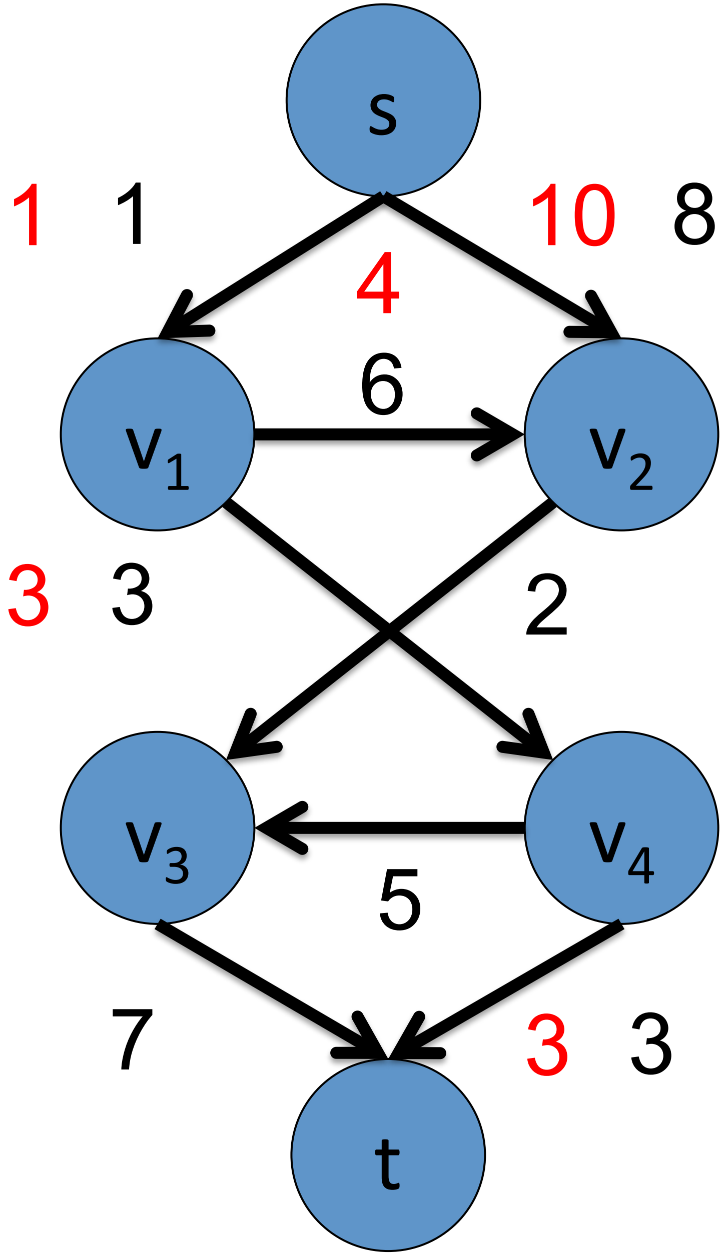
Excess function $E_f(U)$

Incoming Value

-

Outgoing Value

Excess Functions of Vertex Subsets



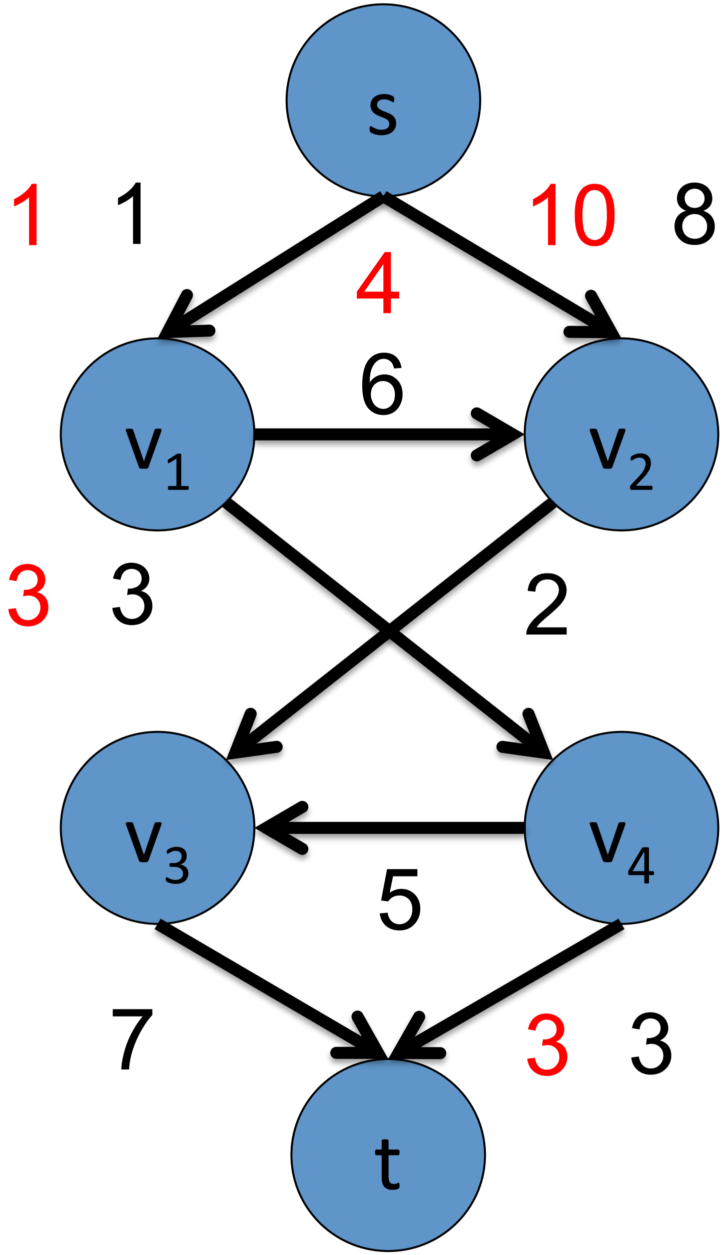
Excess function $E_f(U)$

$\Sigma_{a \in \text{in-arcs}(U)} f(a)$

-

Outgoing Value

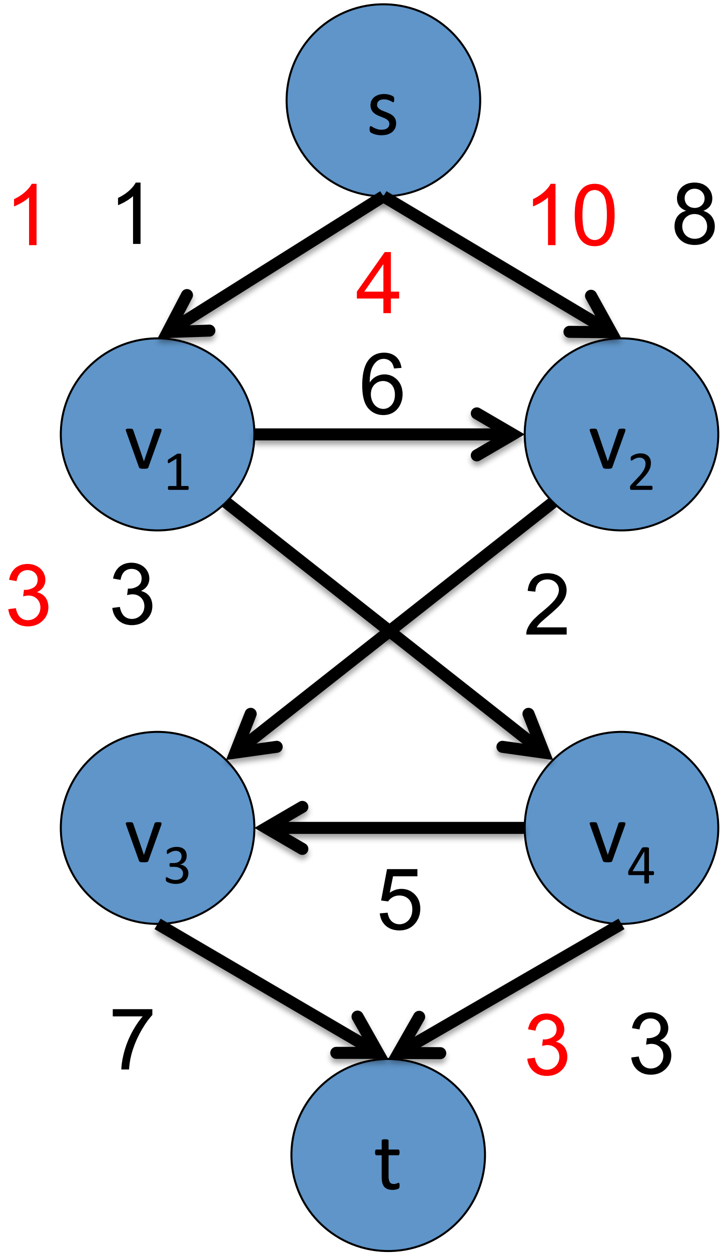
Excess Functions of Vertex Subsets



Excess function $E_f(U)$

$$\Sigma_{a \in \text{in-arcs}(U)} f(a) - \Sigma_{a \in \text{out-arcs}(U)} f(a)$$

Excess Functions of Vertex Subsets



Excess function $E_f(U)$

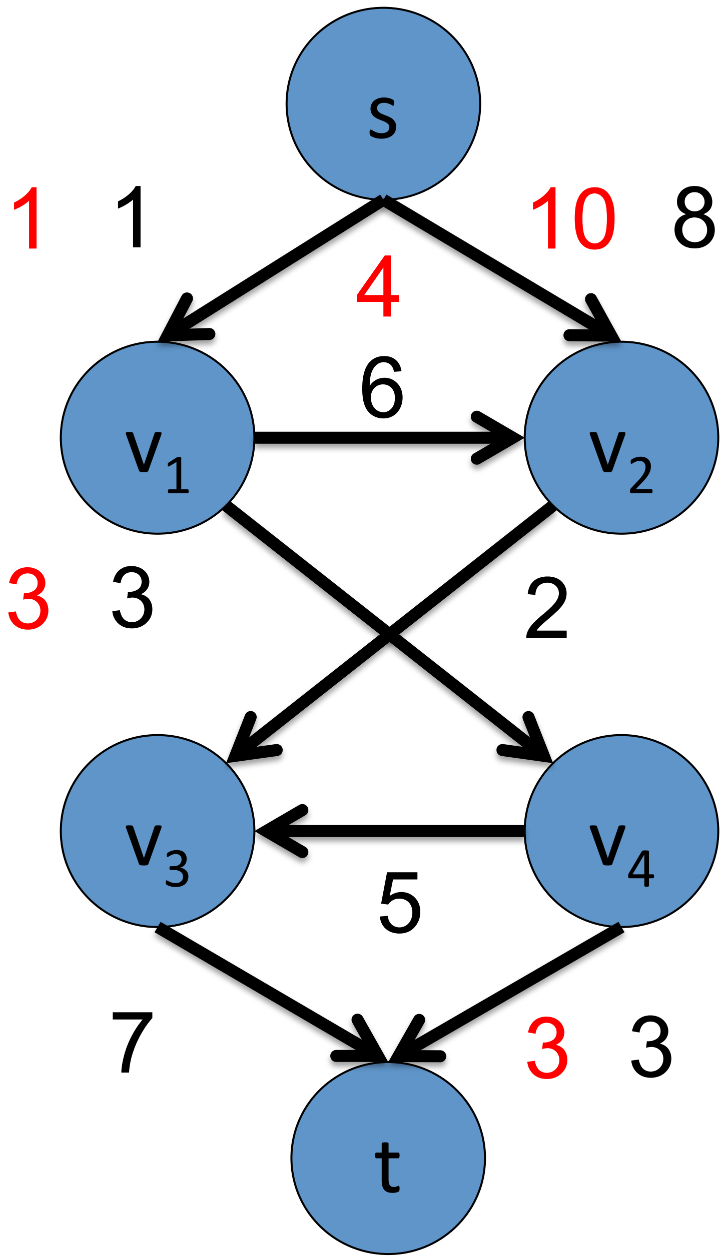
f(in-arcs(U))

-

f(out-arcs(U))

$E_f(\{v_1,v_2\})$ 8

Excess Functions of Vertex Subsets



Excess function $E_f(U)$

f(in-arcs(U))

-

f(out-arcs(U))

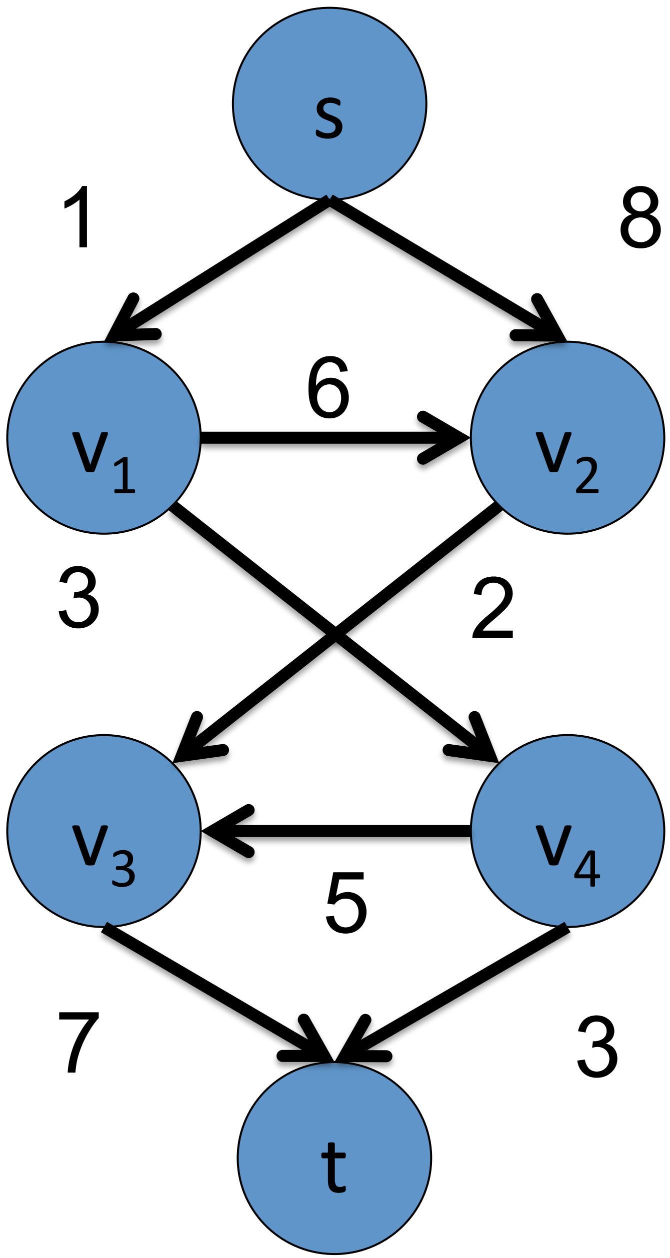
$E_f(\{v_1,v_2\})$ -6 + 14

$E_f(U) = \Sigma_{v\in U} E_f(v)$

Outline

- Preliminaries
  - Functions and Excess Functions
  - **s-t Flow**
  - s-t Cut
  - Flows vs. Cuts
- Maximum Flow
- Algorithms

s-t Flow



Function flow: A ➔ R

Flow of arc ≤ arc capacity

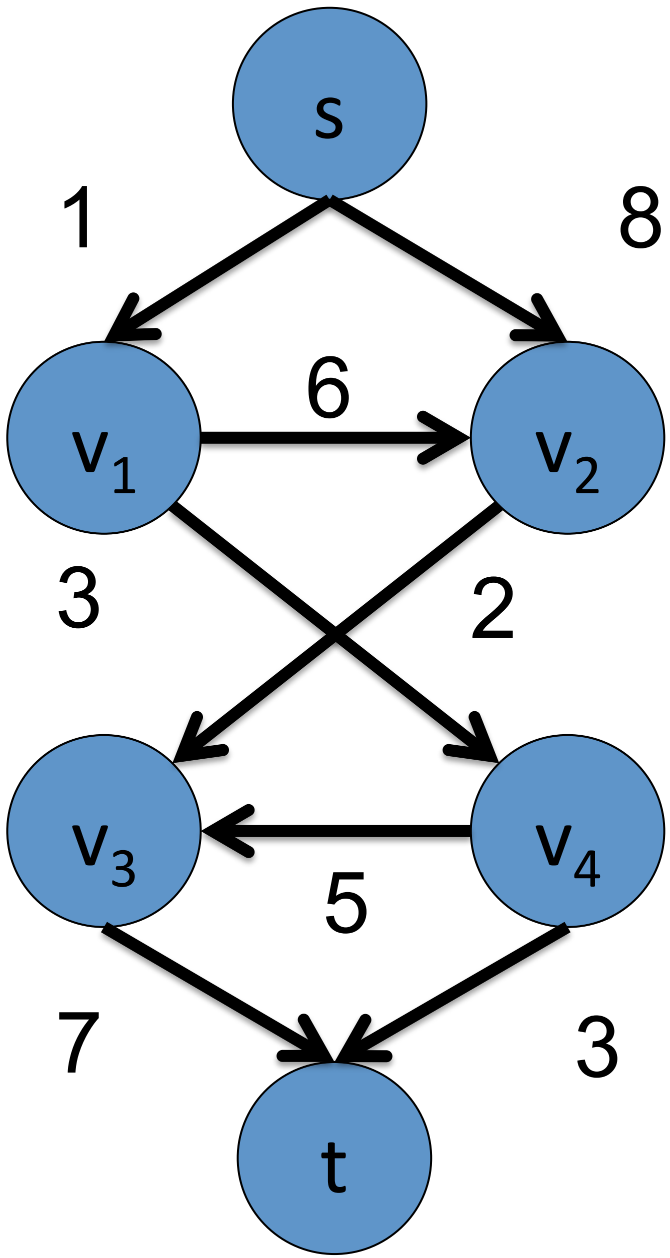
Flow is non-negative

For all vertex expect s,t

Incoming flow

= Outgoing flow

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

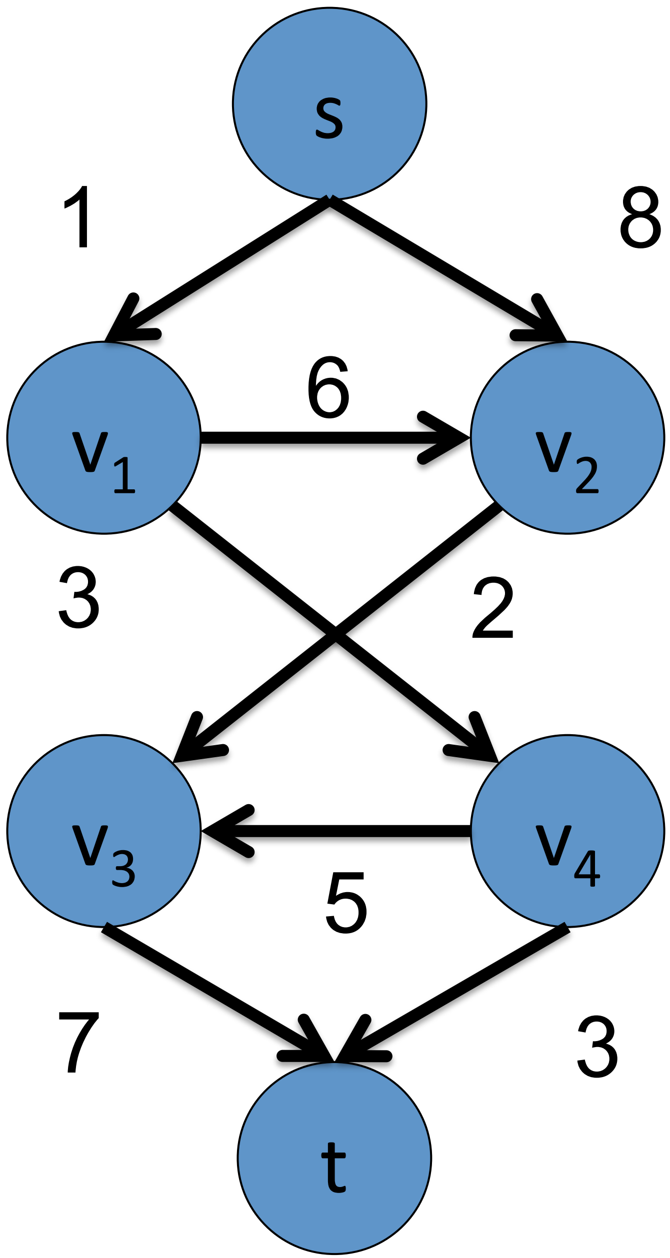
Flow is non-negative

For all vertex expect s,t

Incoming flow

= Outgoing flow

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

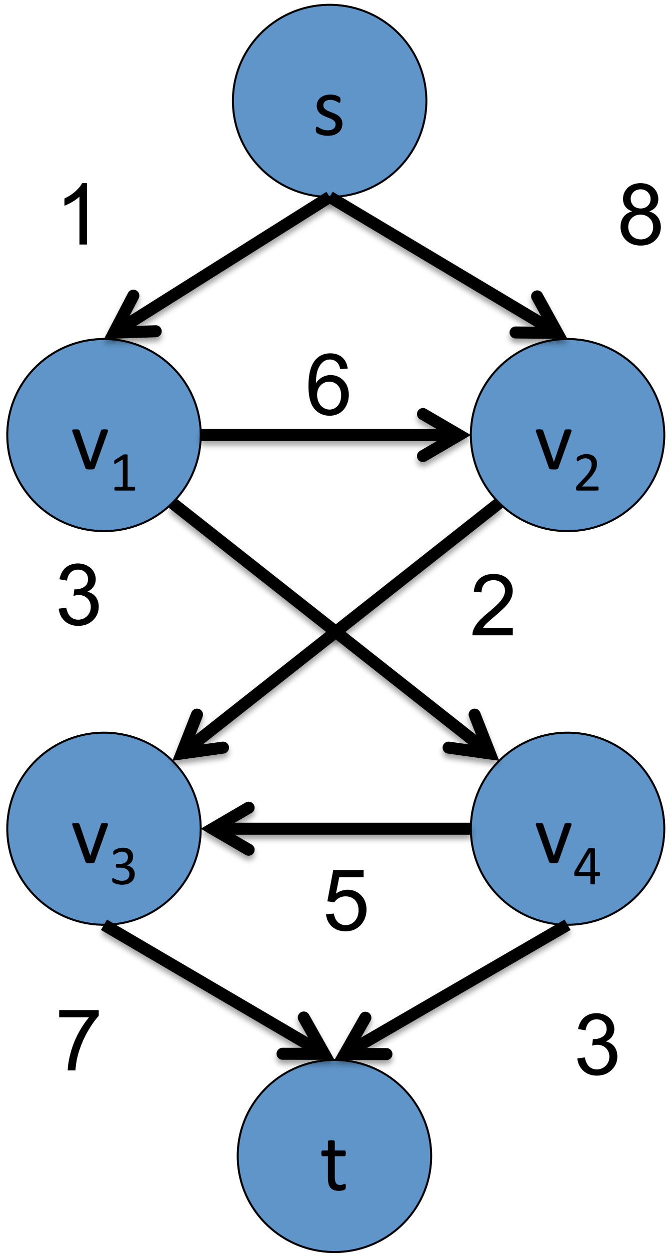
flow(a) ≥ 0

For all vertex expect s,t

Incoming flow

= Outgoing flow

s-t Flow



Function flow: A → R

flow(a) ≤ c(a)

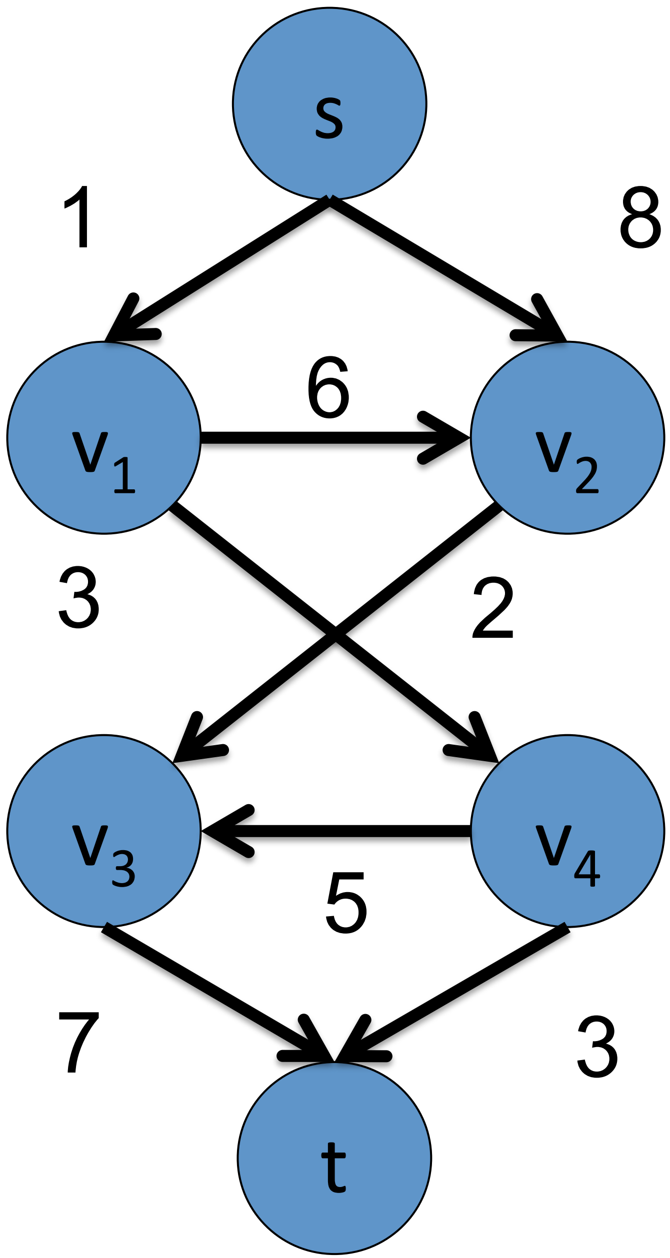
flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

Incoming flow

= Outgoing flow

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

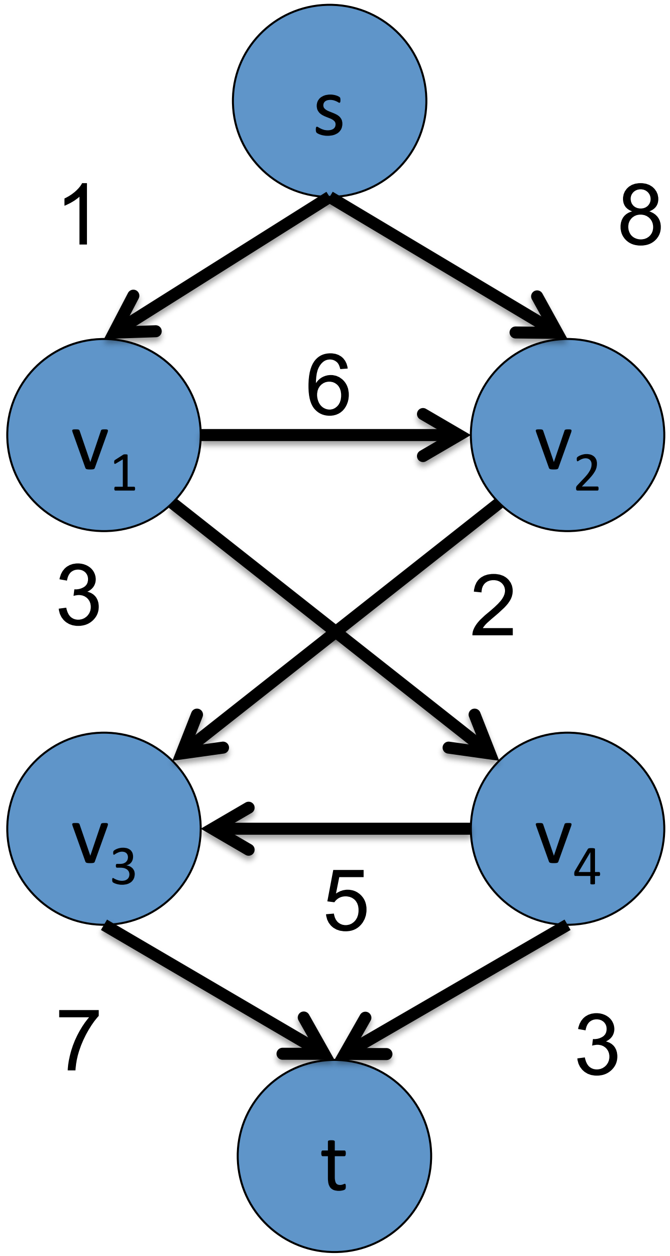
flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

$\Sigma_{(u,v)\in A}$ flow((u,v))

= Outgoing flow

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

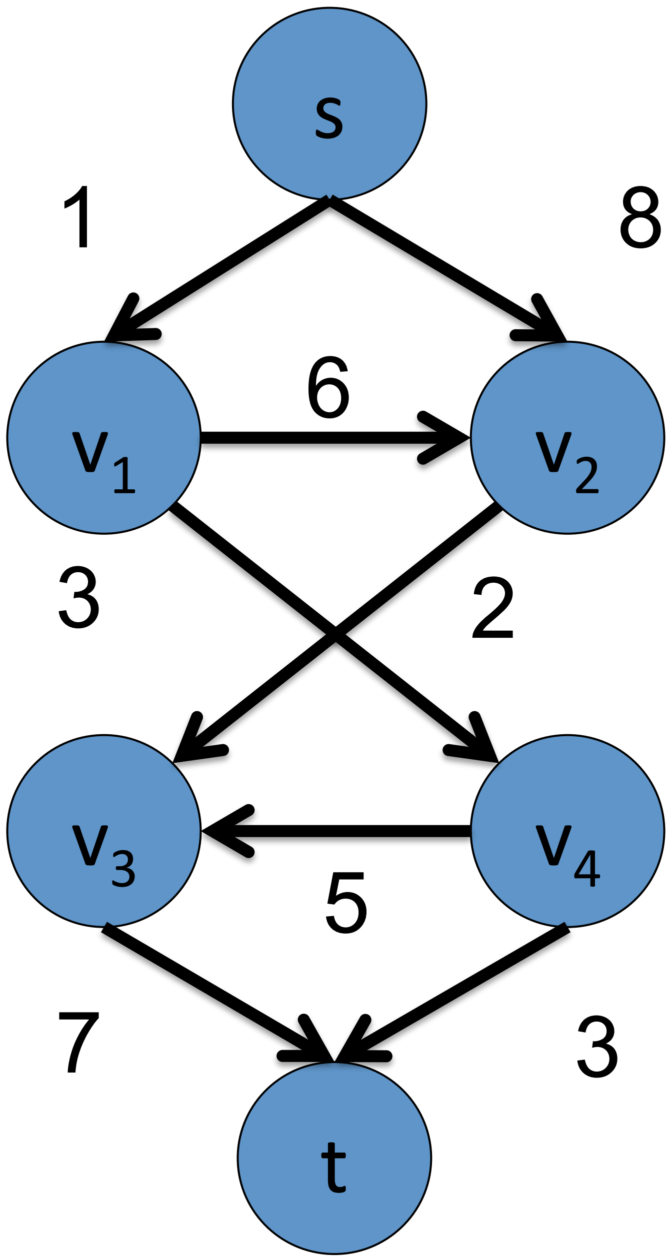
flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

$\Sigma_{(u,v)\in A}$ flow((u,v))

$= \Sigma_{(v,u)\in A}$ flow((v,u))

s-t Flow



Function flow: A → R

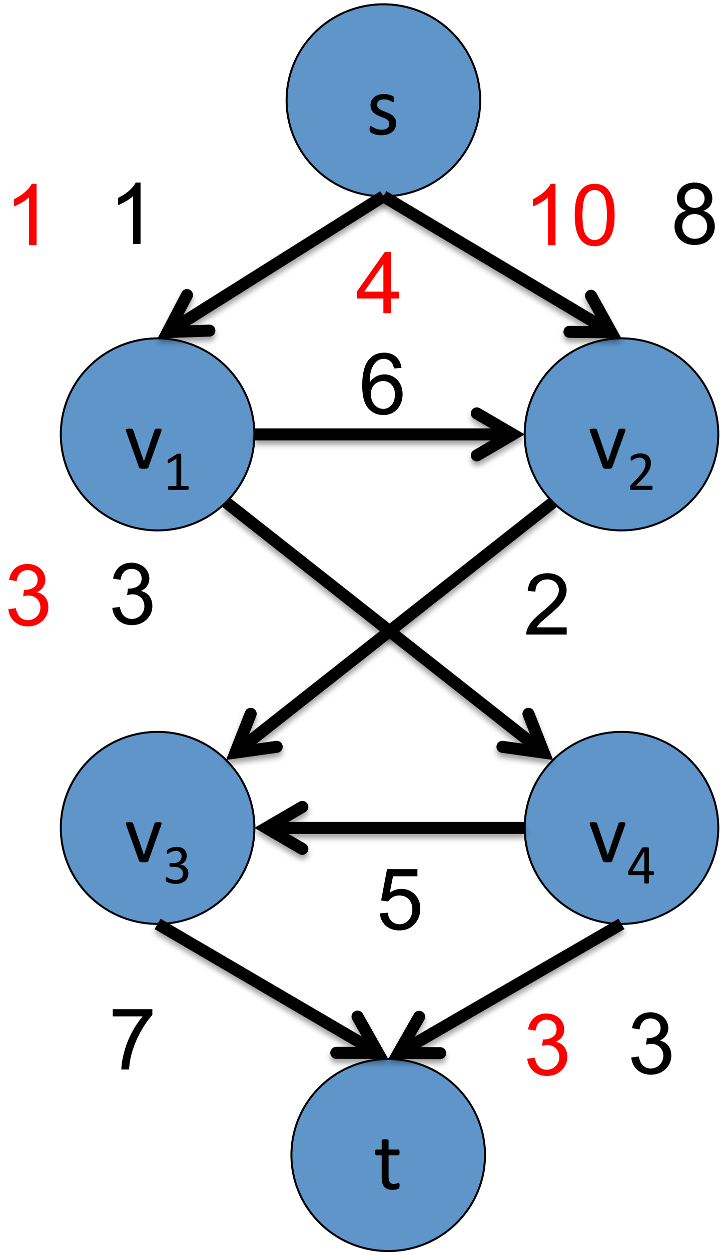
flow(a) ≤ c(a)

flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

$E_{flow}(v) = 0$

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

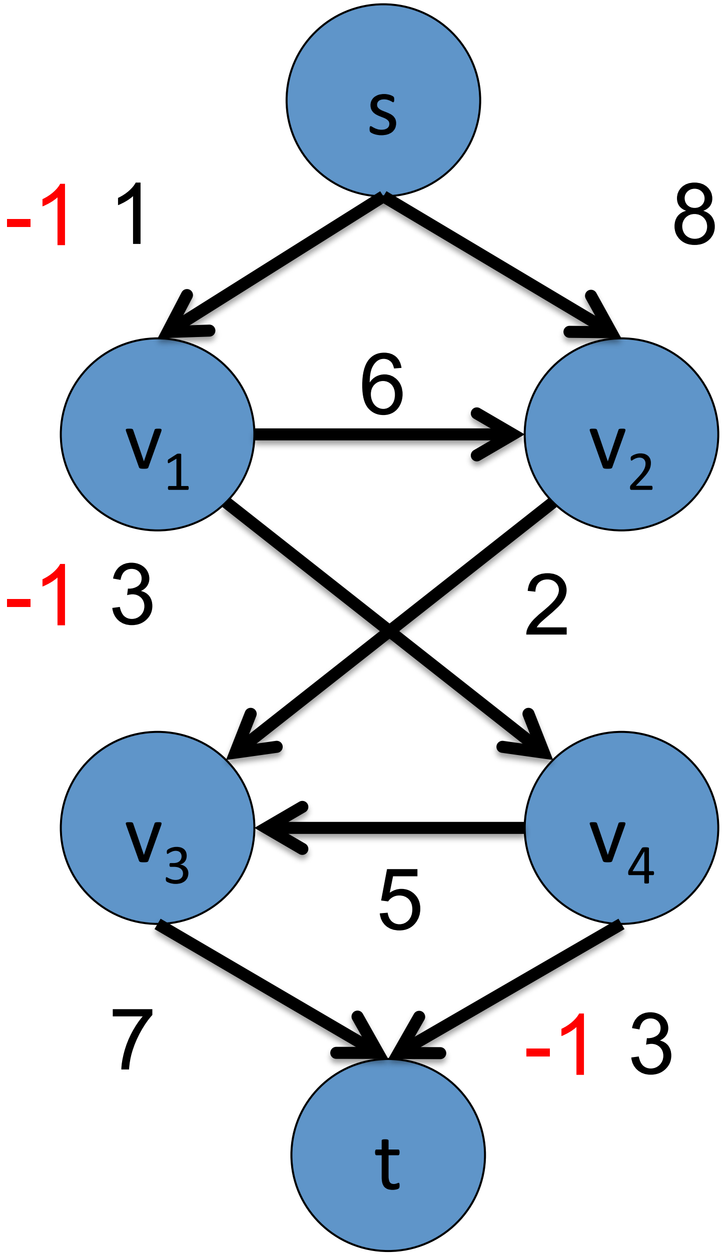
flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

$E_{flow}(v) = 0$

✗

s-t Flow



Function flow: A ➔ R

flow(a) ≤ c(a)

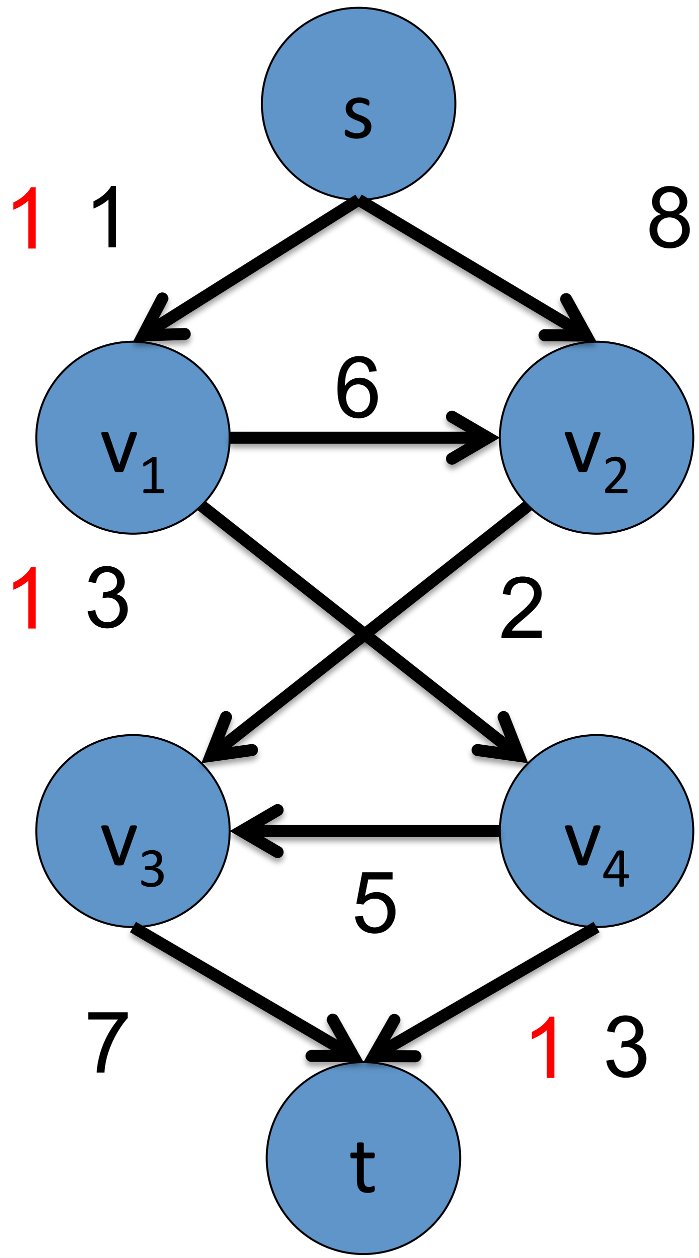
flow(a) ≥ 0

For all v ∈ V \ {s,t}

$E_{flow}(v) = 0$

✗

s-t Flow



Function flow: A → R

flow(a) ≤ c(a)

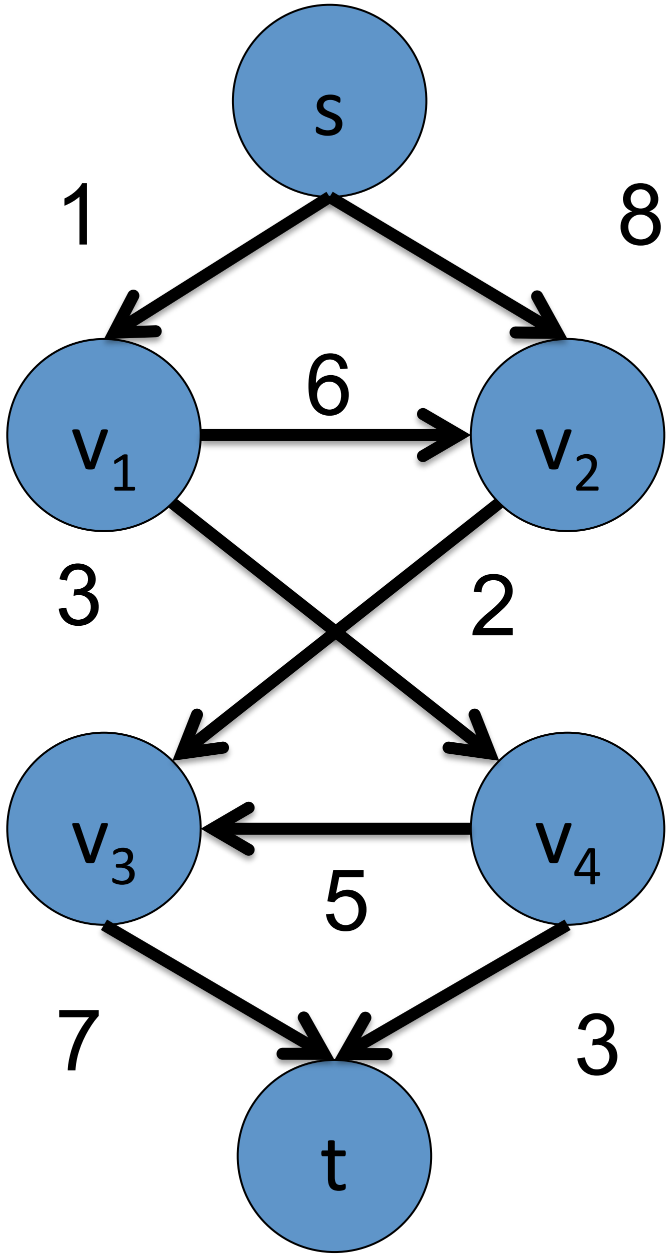
flow(a) ≥ 0

For all $v \in V \setminus \{s,t\}$

$E_{flow}(v) = 0$

✓

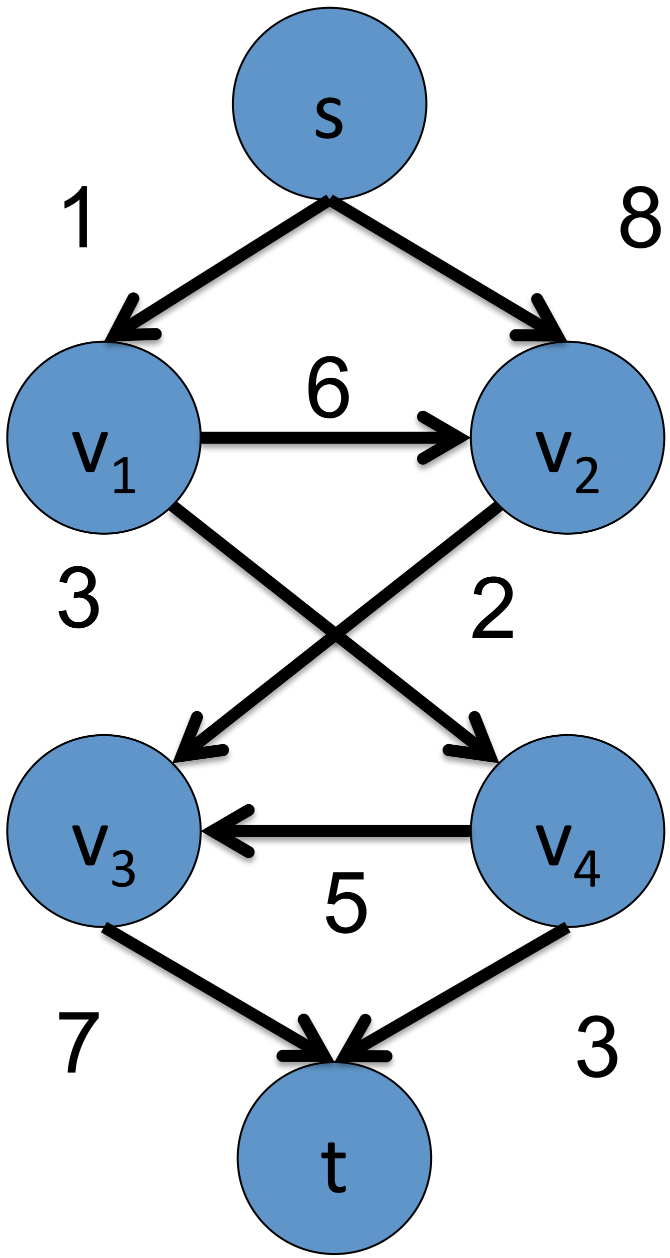
Value of s-t Flow



Outgoing flow of s

- Incoming flow of s

Value of s-t Flow

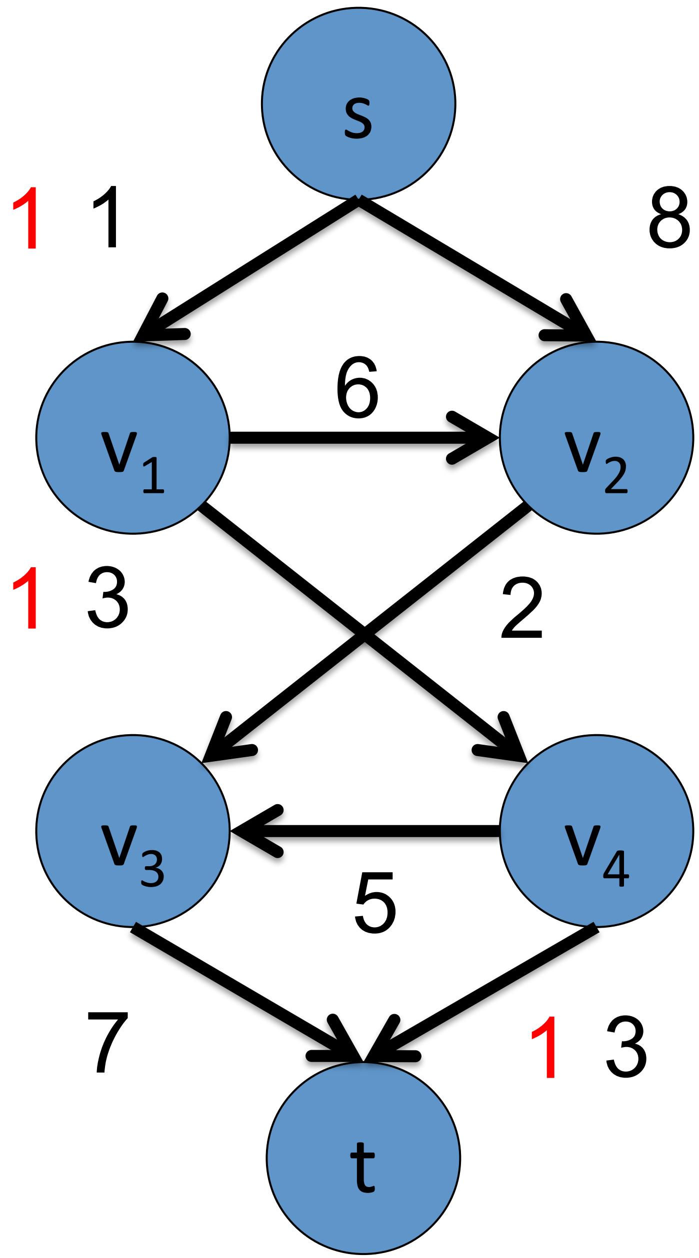


$-E_{flow}(s)$ $\quad$ $E_{flow}(t)$

$\Sigma_{(s,v)\in A}$ flow((s,v))

$-\ \Sigma_{(u,s)\in A}$ flow((u,s))

Value of s-t Flow



$-E_{flow}(s)$ $E_{flow}(t)$

$\Sigma_{(s,v)\in A}$ flow((s,v))

$- \Sigma_{(u,s)\in A}$ flow((u,s))

Value = 1

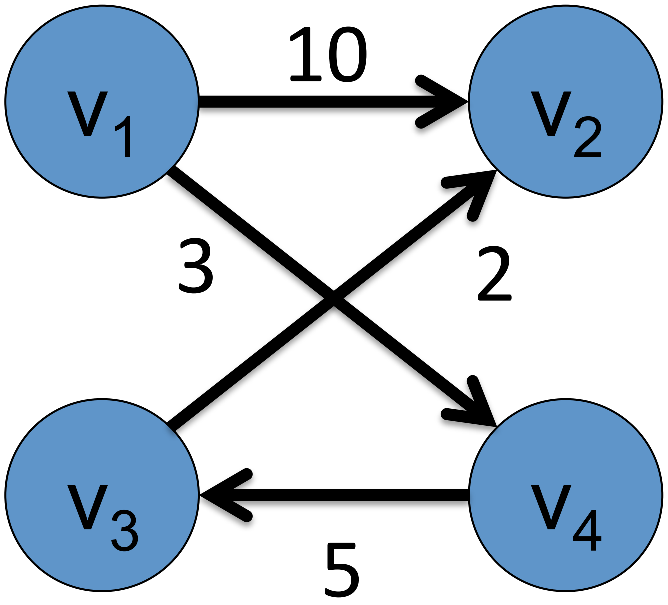
Outline

- Preliminaries
  - Functions and Excess Functions
  - s-t Flow
  - **s-t Cut**
  - Flows vs. Cuts
- Maximum Flow
- Algorithms

Cut

D = (V, A)

Let U be a subset of V

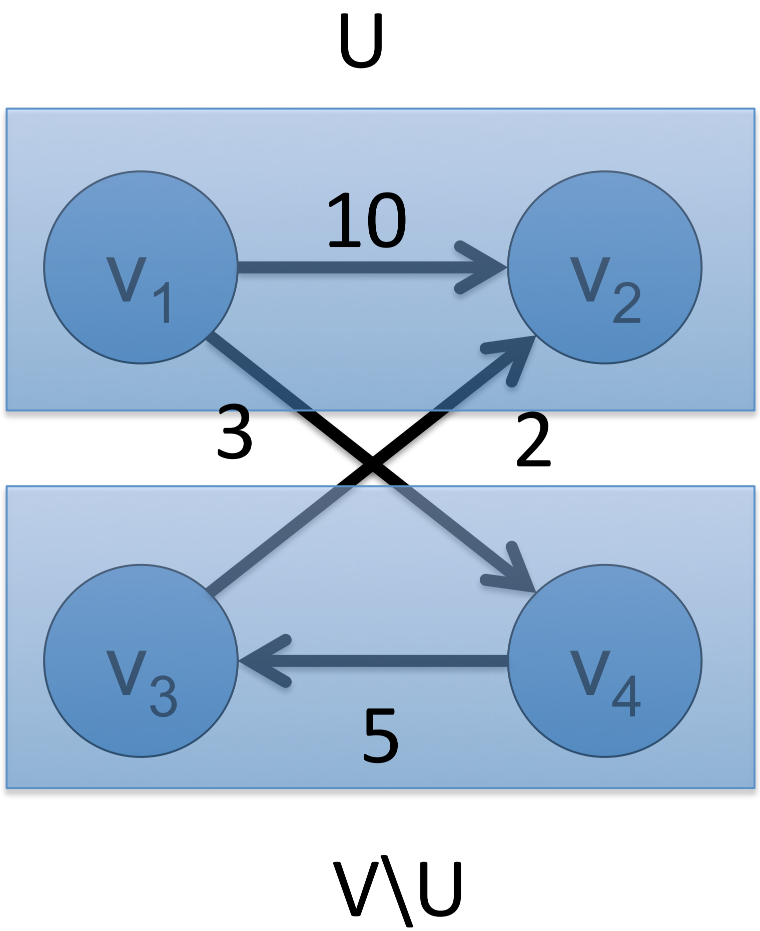


C is a set of arcs such that

- $(u,v) \in A$
- $u \in U$
- $v \in V \backslash U$

C is a cut in the digraph D

Cut



D = (V, A)

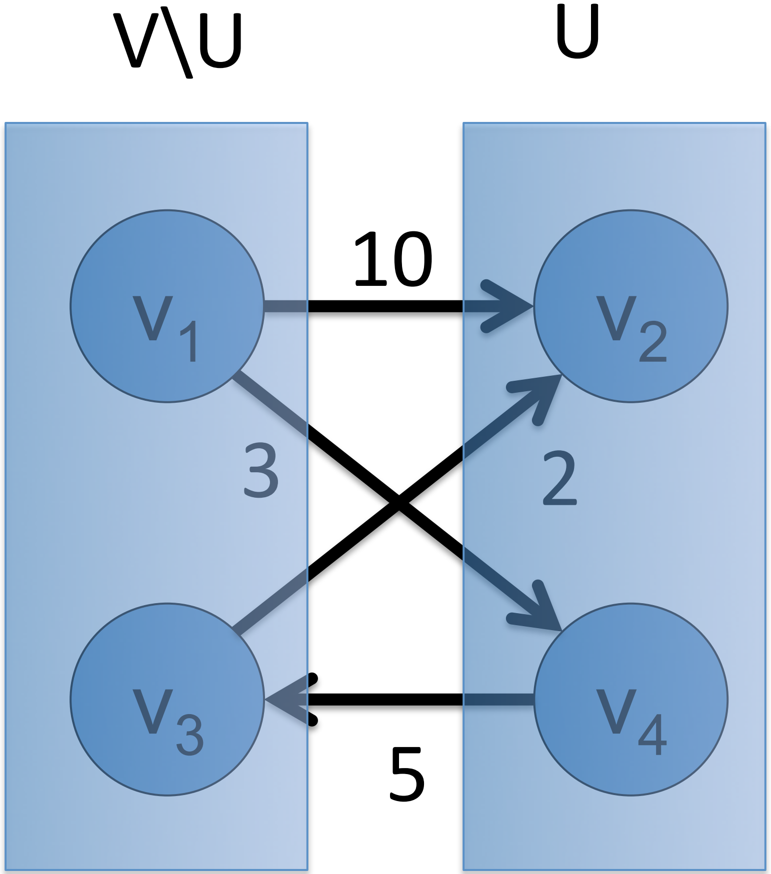
What is C?

$\{(v_1,v_2),(v_1,v_4)\}$?

$\{(v_1,v_4),(v_3,v_2)\}$?

✓ $\{(v_1,v_4)\}$?

Cut



D = (V, A)

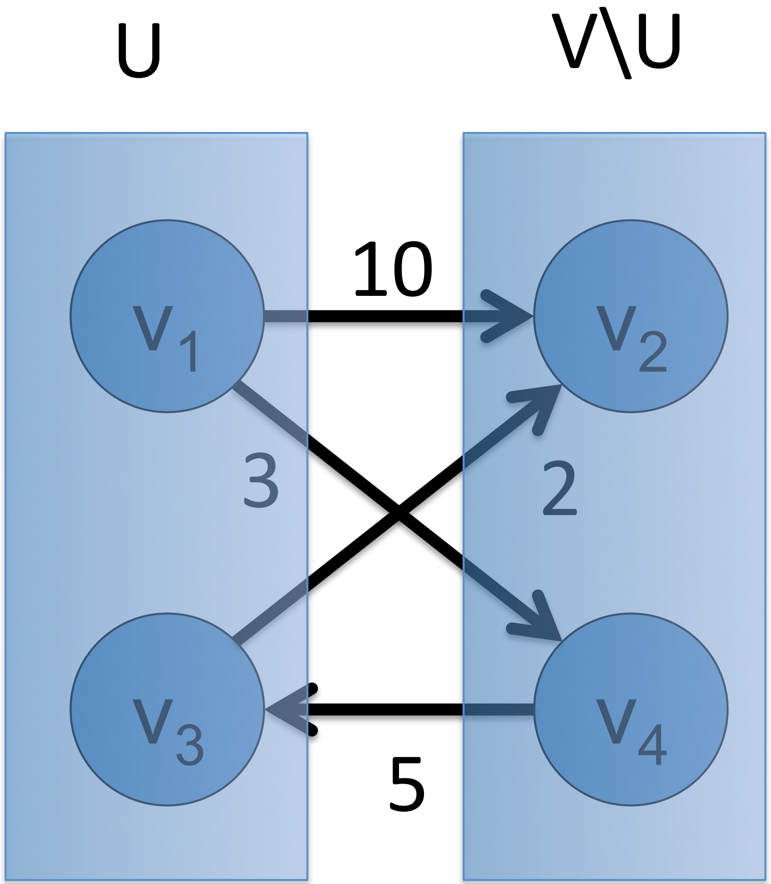
What is C?

$\{(v_1,v_2),(v_1,v_4),(v_3,v_2)\}$?

✓ $\{(v_4,v_3)\}$?

$\{(v_1,v_4),(v_3,v_2)\}$?

Cut



D = (V, A)

What is C?

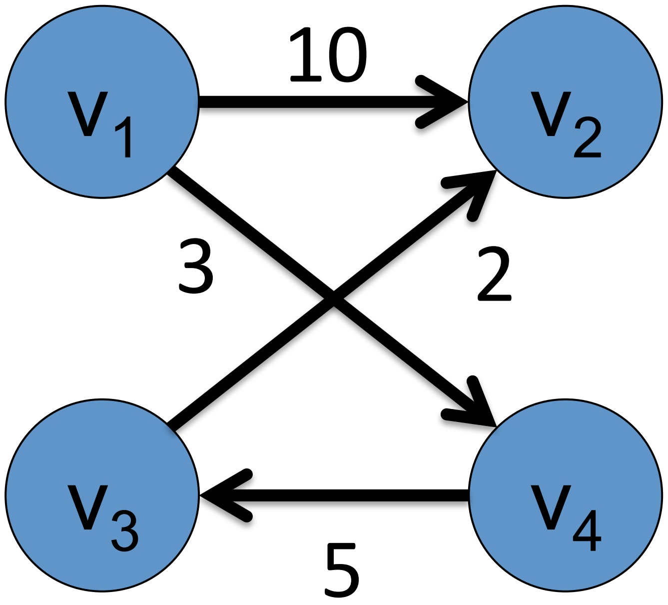
✓ $\{(v_1,v_2),(v_1,v_4),(v_3,v_2)\}$?

$\{(v_3,v_2)\}$?

$\{(v_1,v_4),(v_3,v_2)\}$?

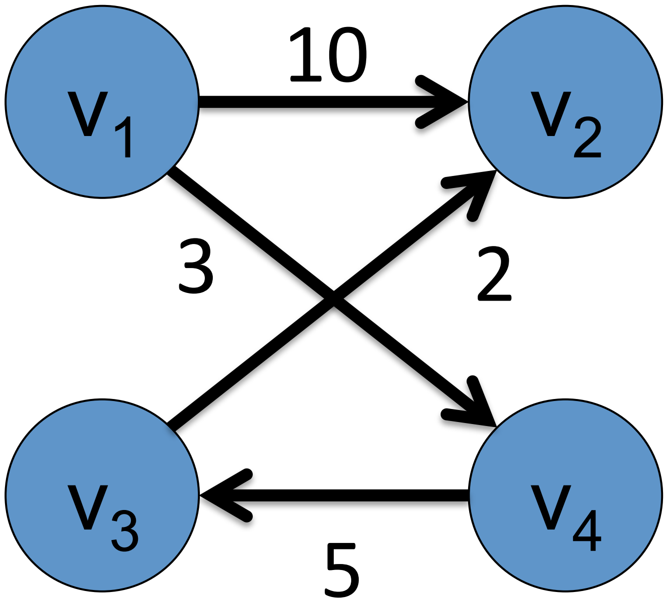
Cut

D = (V, A)



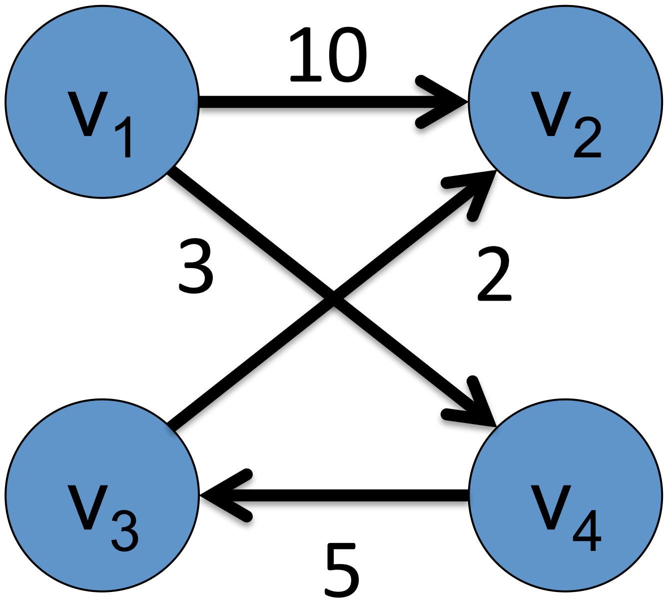
C = out-arcs(U)

Capacity of Cut



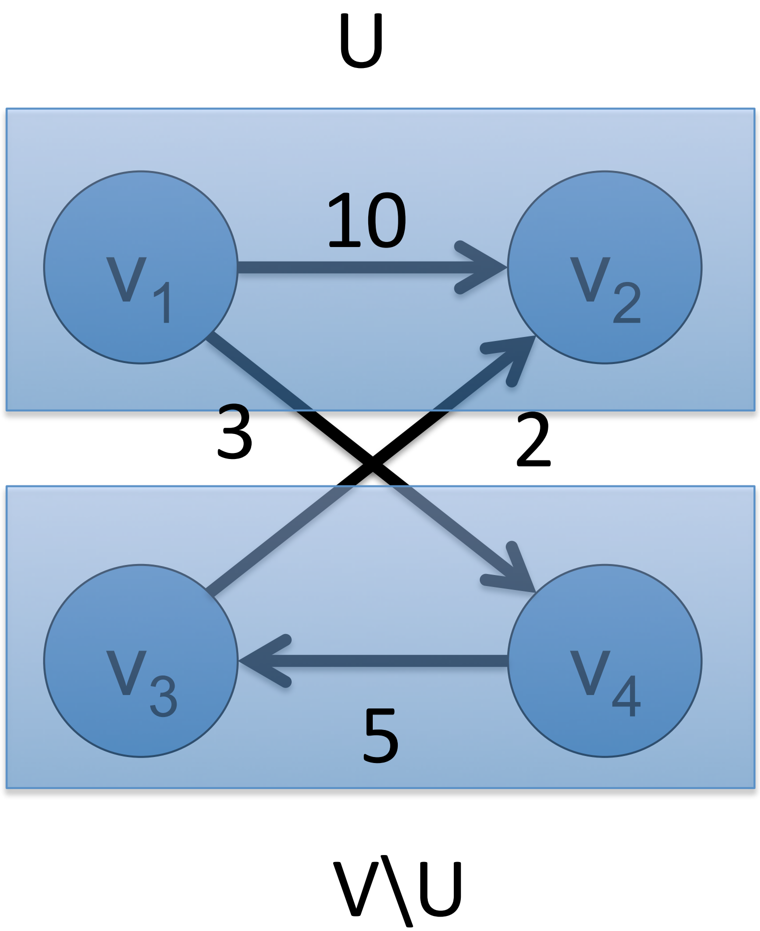
Sum of capacity of all arcs in C

Capacity of Cut



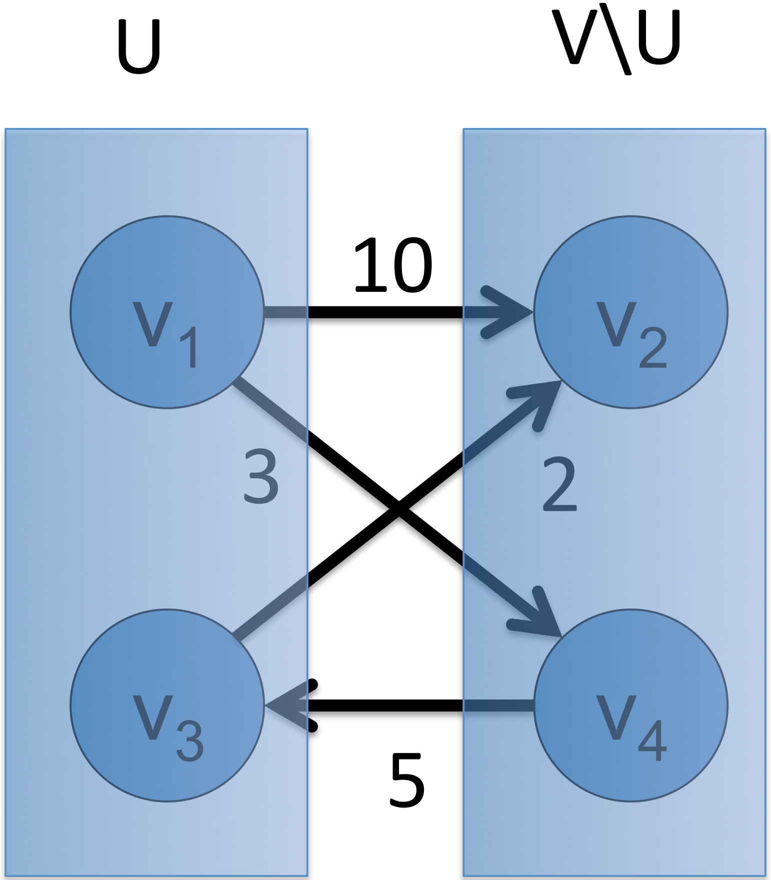
$\Sigma_{a \in C}\, c(a)$

Capacity of Cut



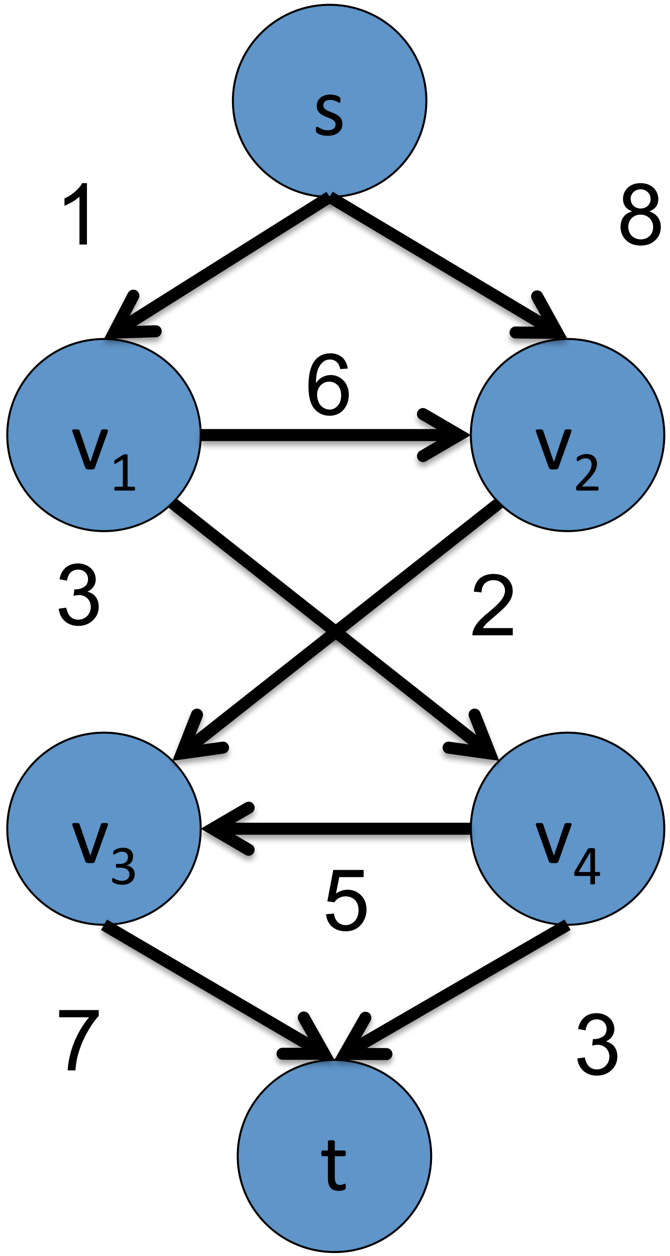
3

Capacity of Cut



15

s-t Cut



D = (V, A)

A source vertex "s"

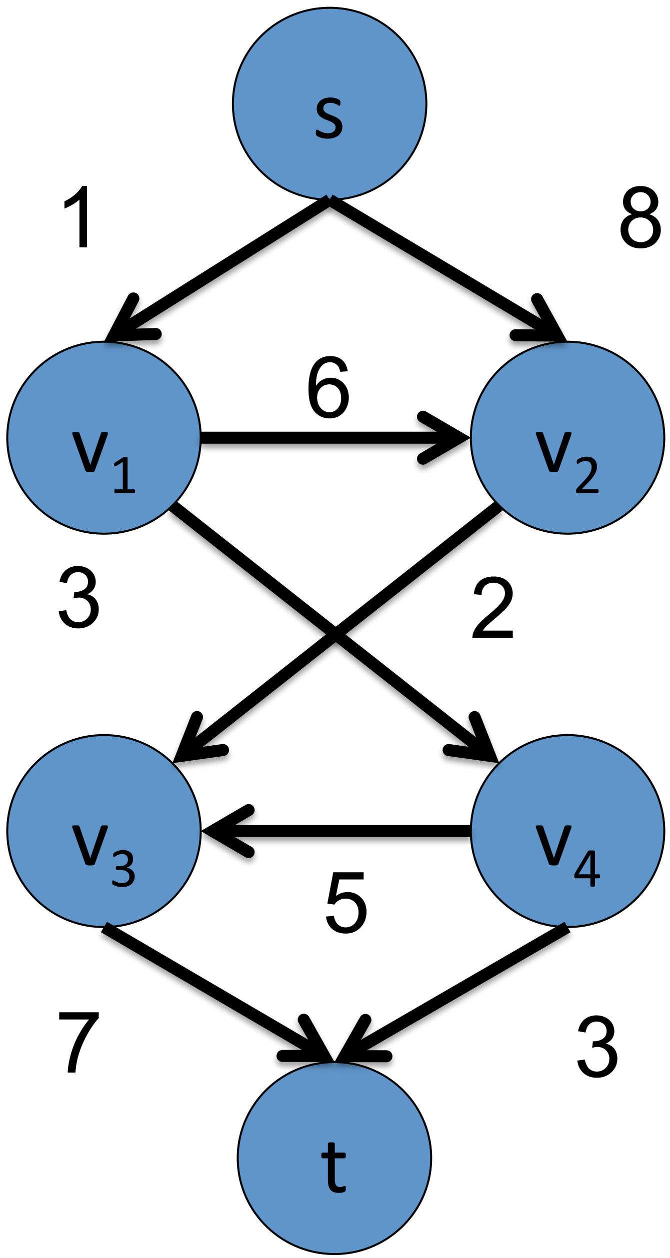
A sink vertex "t"

C is a cut such that

- $s \in U$
- $t \in V \backslash U$

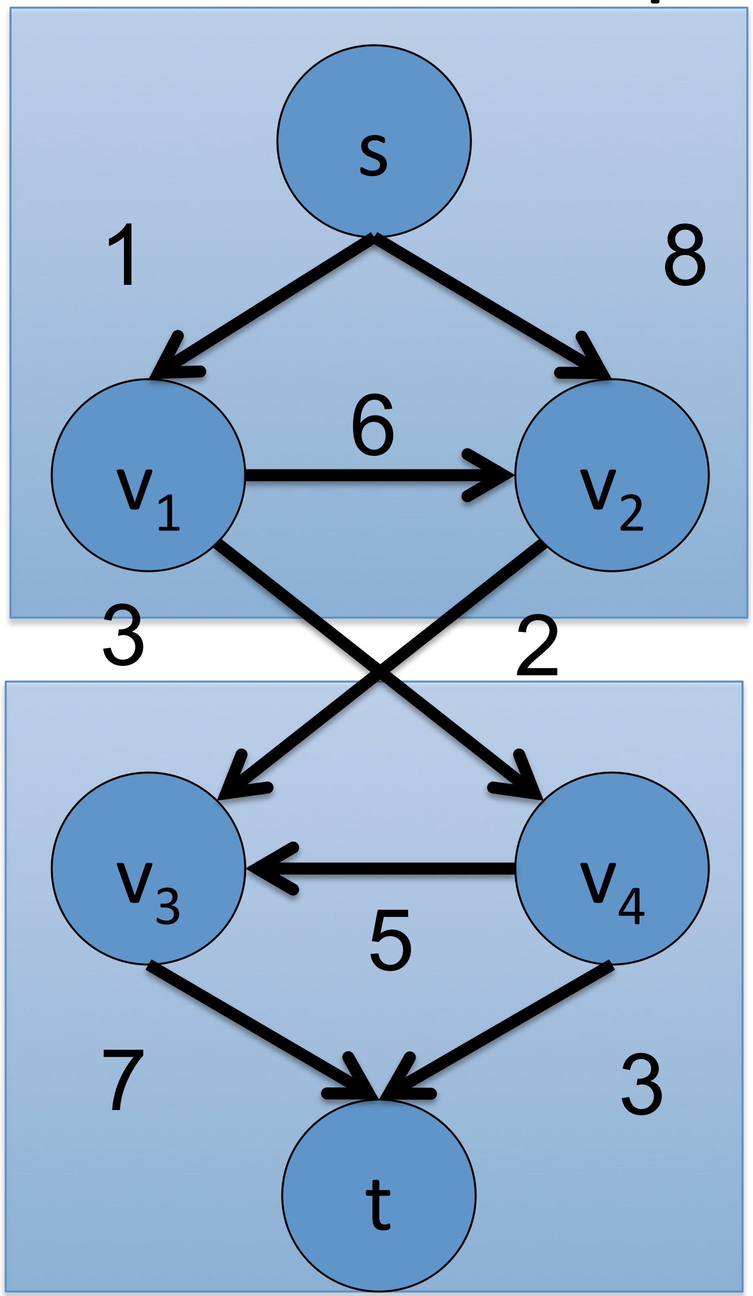
C is an s-t cut

Capacity of s-t Cut



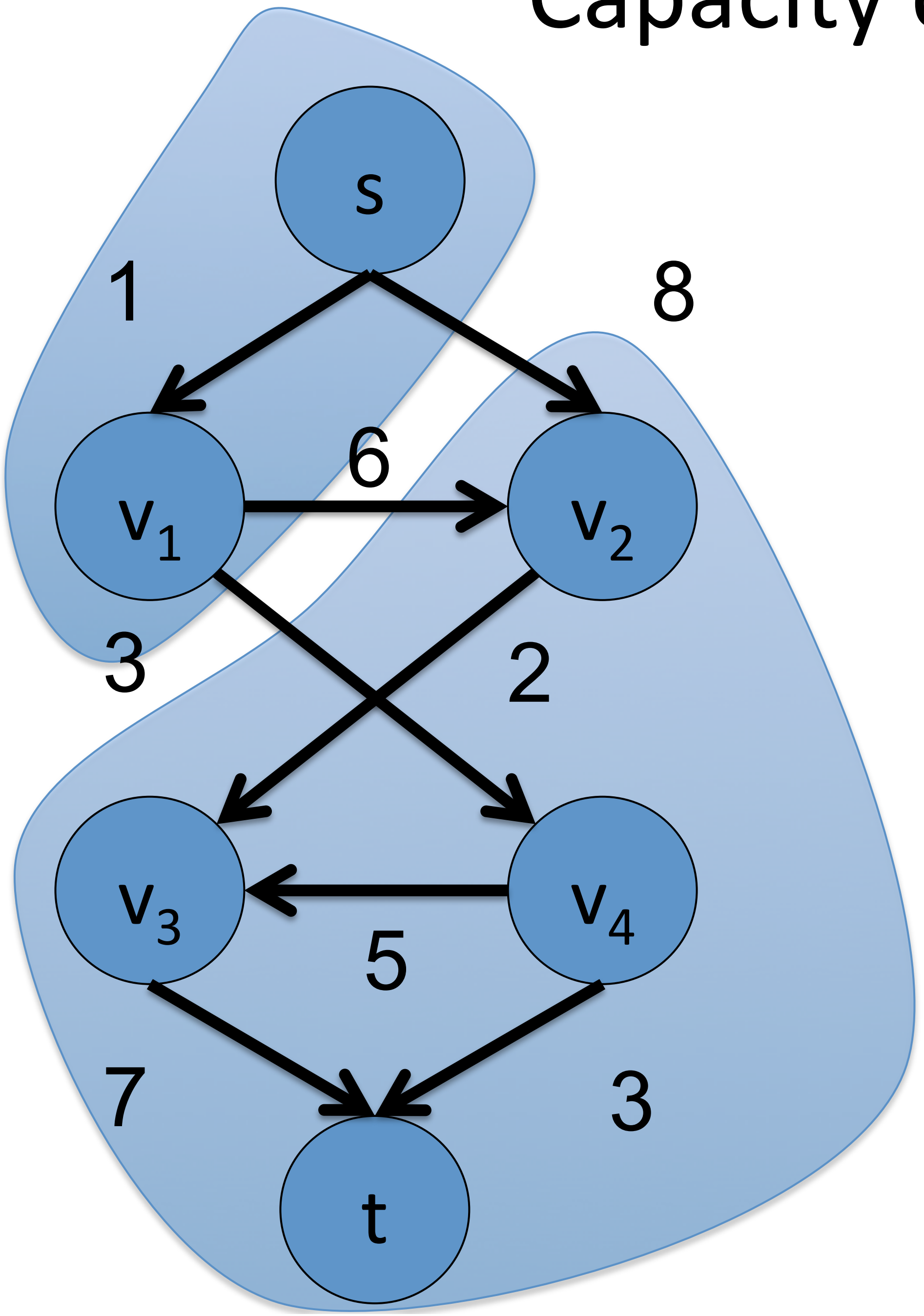
$\Sigma_{a \in C}\, c(a)$

Capacity of s-t Cut



5

Capacity of s-t Cut



17

Outline

- Preliminaries
  - Functions and Excess Functions
  - s-t Flow
  - s-t Cut
  - **Flows vs. Cuts**
- Maximum Flow
- Algorithms

Flows vs. Cuts

An s-t flow function flow: A ➔ Reals

An s-t cut C such that $s \in U$, $t \in V \setminus U$

Value of flow ≤ Capacity of C

Flows vs. Cuts

Value of flow $= -E_{flow}(s)$

$= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$

$= -E_{flow}(U)$

= flow(out-arcs(U))
 - flow(in-arcs(U))

≤ Capacity of C
 - flow(in-arcs(U))

Flows vs. Cuts

Value of flow $= -E_{flow}(s)$

$= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$

$= -E_{flow}(U)$

= flow(out-arcs(U))
 - flow(in-arcs(U))

≤ Capacity of C

When does equality hold?

Flows vs. Cuts

Value of flow $= -E_{flow}(s)$

$= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$

$= -E_{flow}(U)$

$=$ flow(out-arcs(U))

$-$ flow(in-arcs(U))

$\leq$ Capacity of C

flow(a) = c(a), a $\in$ out-arcs(U)

flow(a) = 0, a $\in$ in-arcs(U)

Flows vs. Cuts

Value of flow $= -E_{flow}(s)$

$= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$

$= -E_{flow}(U)$

= flow(out-arcs(U))
 - flow(in-arcs(U))

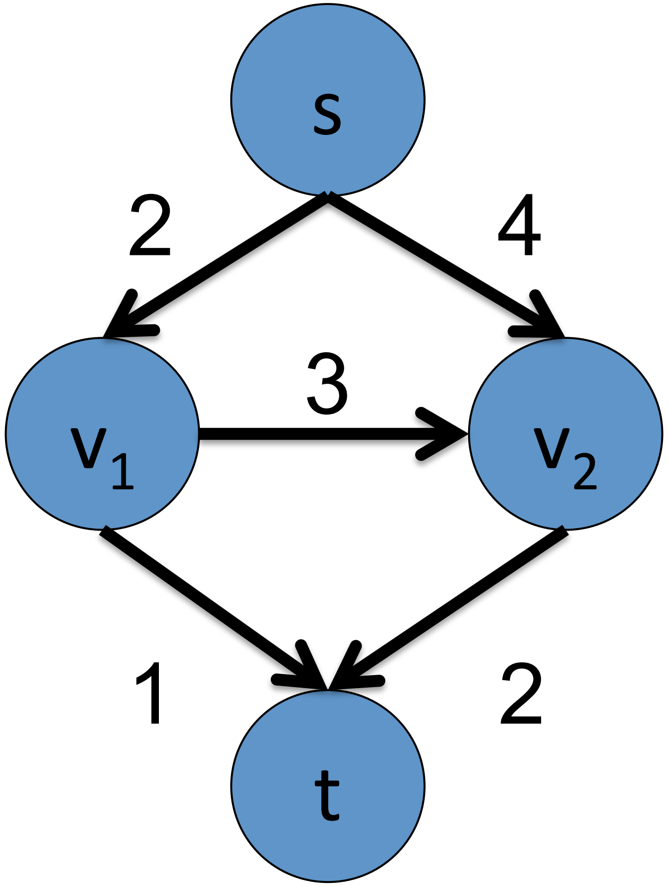
= Capacity of C

flow(a) = c(a), $a \in$ out-arcs(U)   flow(a) = 0, $a \in$ in-arcs(U)

Outline

- Preliminaries
- **Maximum Flow**
  - Residual Graph
  - Max-Flow Min-Cut Theorem
- Algorithms

Maximum Flow Problem



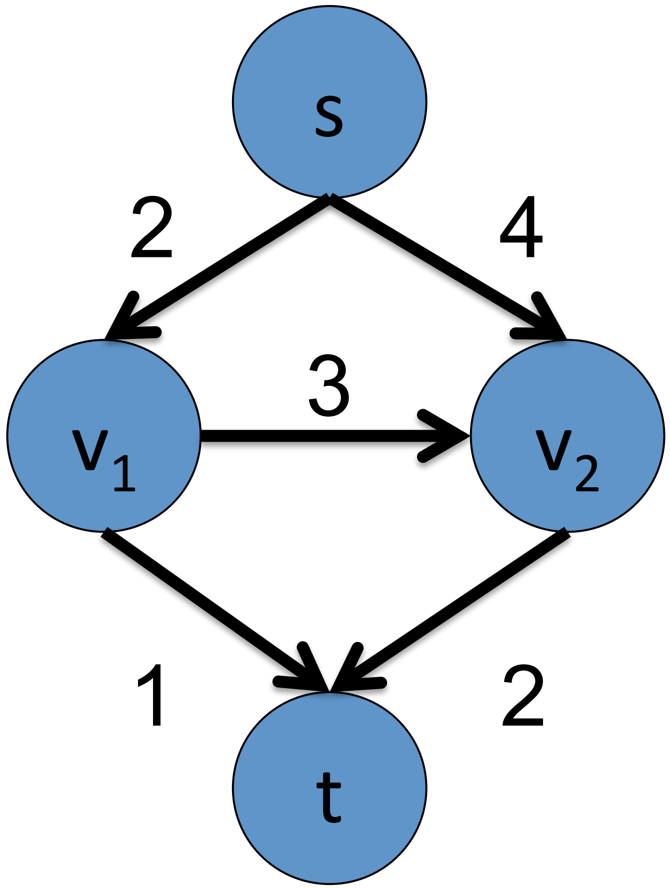
Find the flow with the maximum value !!

$\Sigma_{(s,v)\in A}$ flow((s,v))

- $\Sigma_{(u,s)\in A}$ flow((u,s))

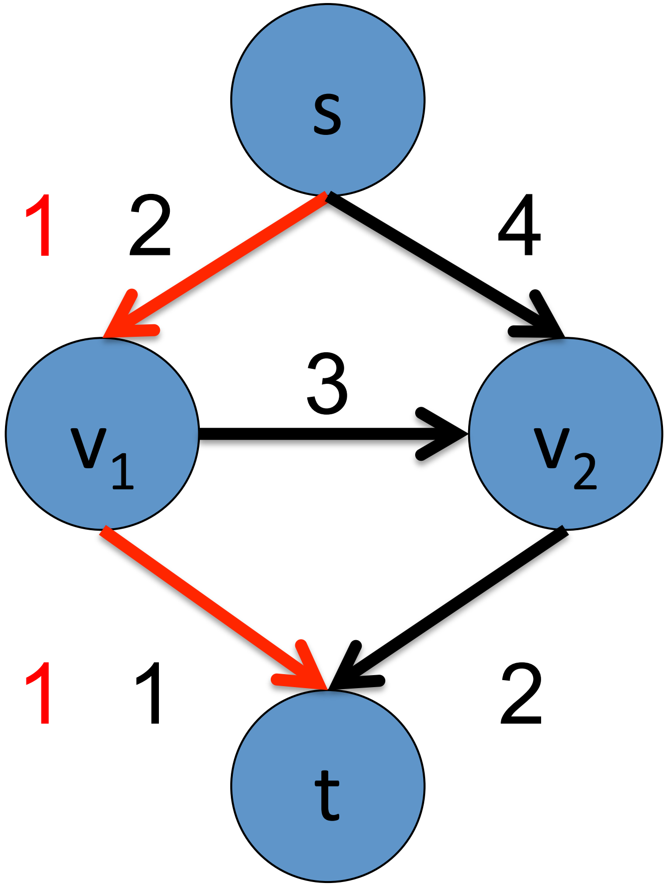
One suggestion to solve this problem !!

Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

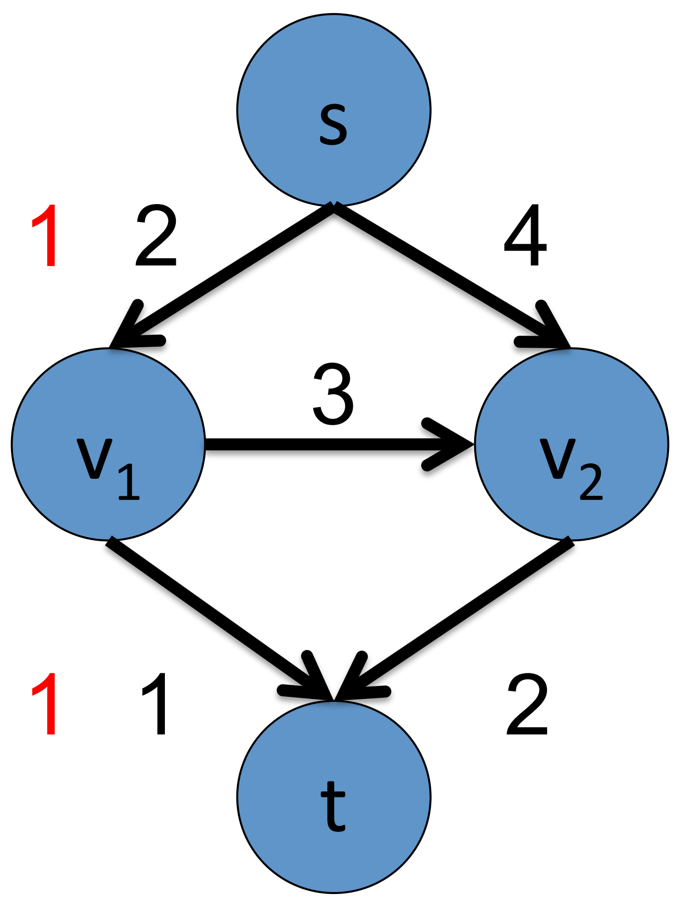
Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

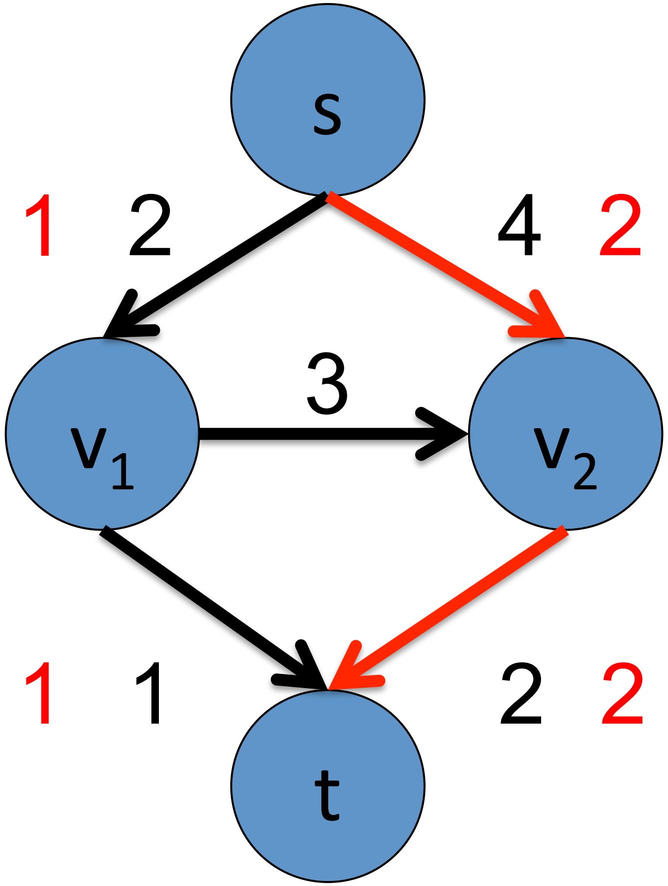
Pass maximum allowable flow through the arcs

Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

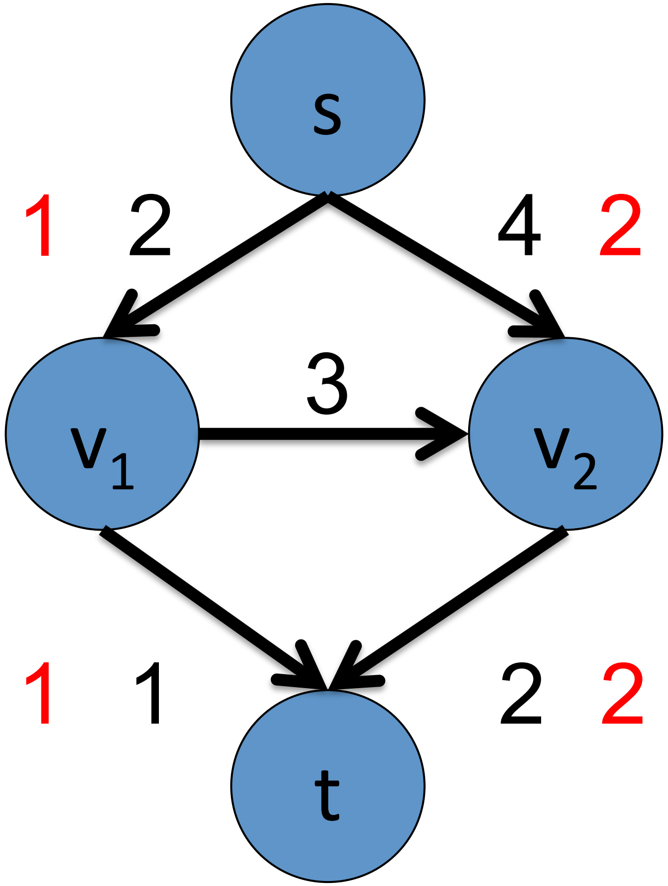
Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs

Passing Flow through s-t Paths

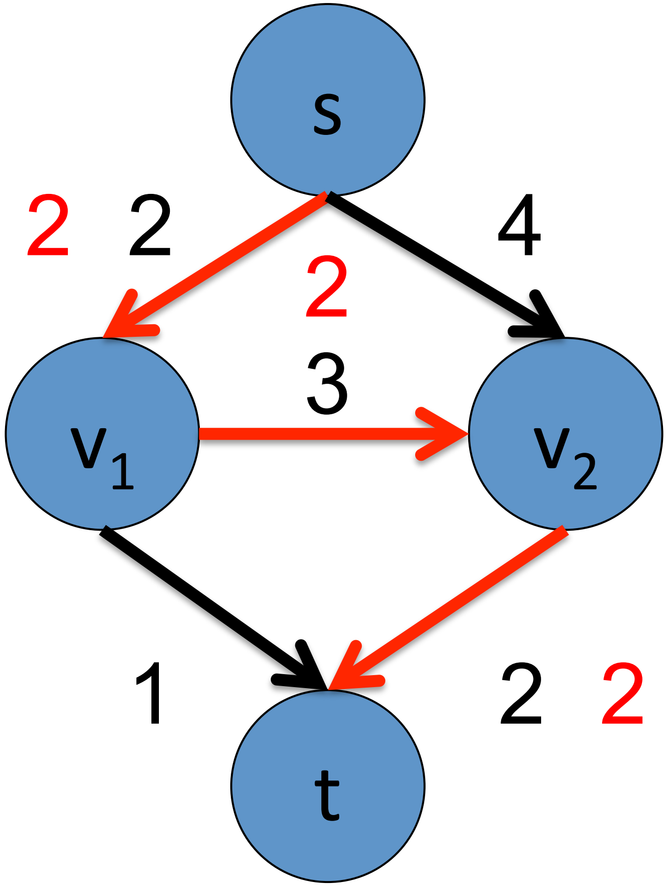


Find an s-t path where flow(a) < c(a) for all arcs

No more paths. Stop.

Will this give us maximum flow? **NO !!!**

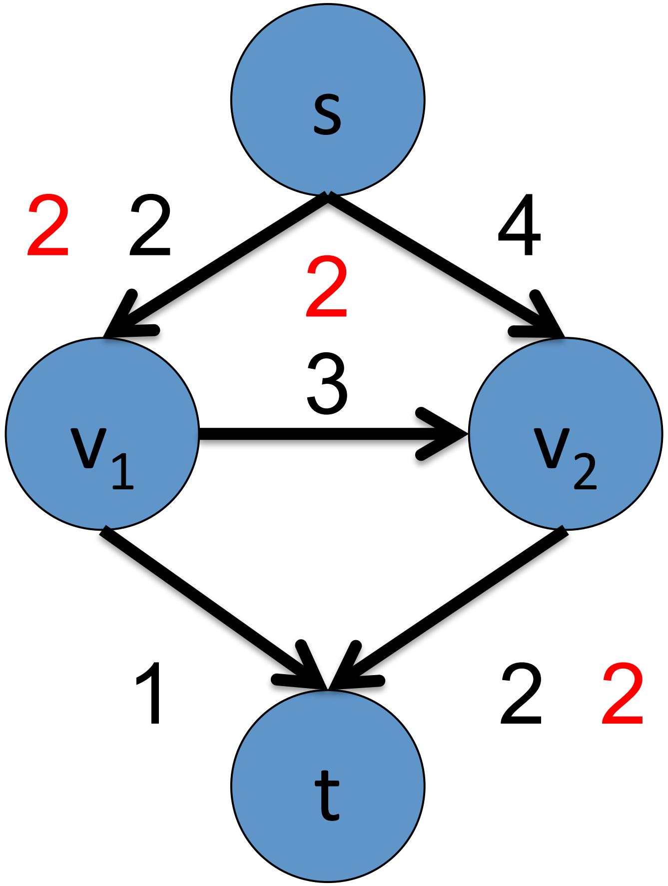
Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs

Passing Flow through s-t Paths



Find an s-t path where flow(a) < c(a) for all arcs

No more paths. Stop.

Another method?

Incorrect Answer !!

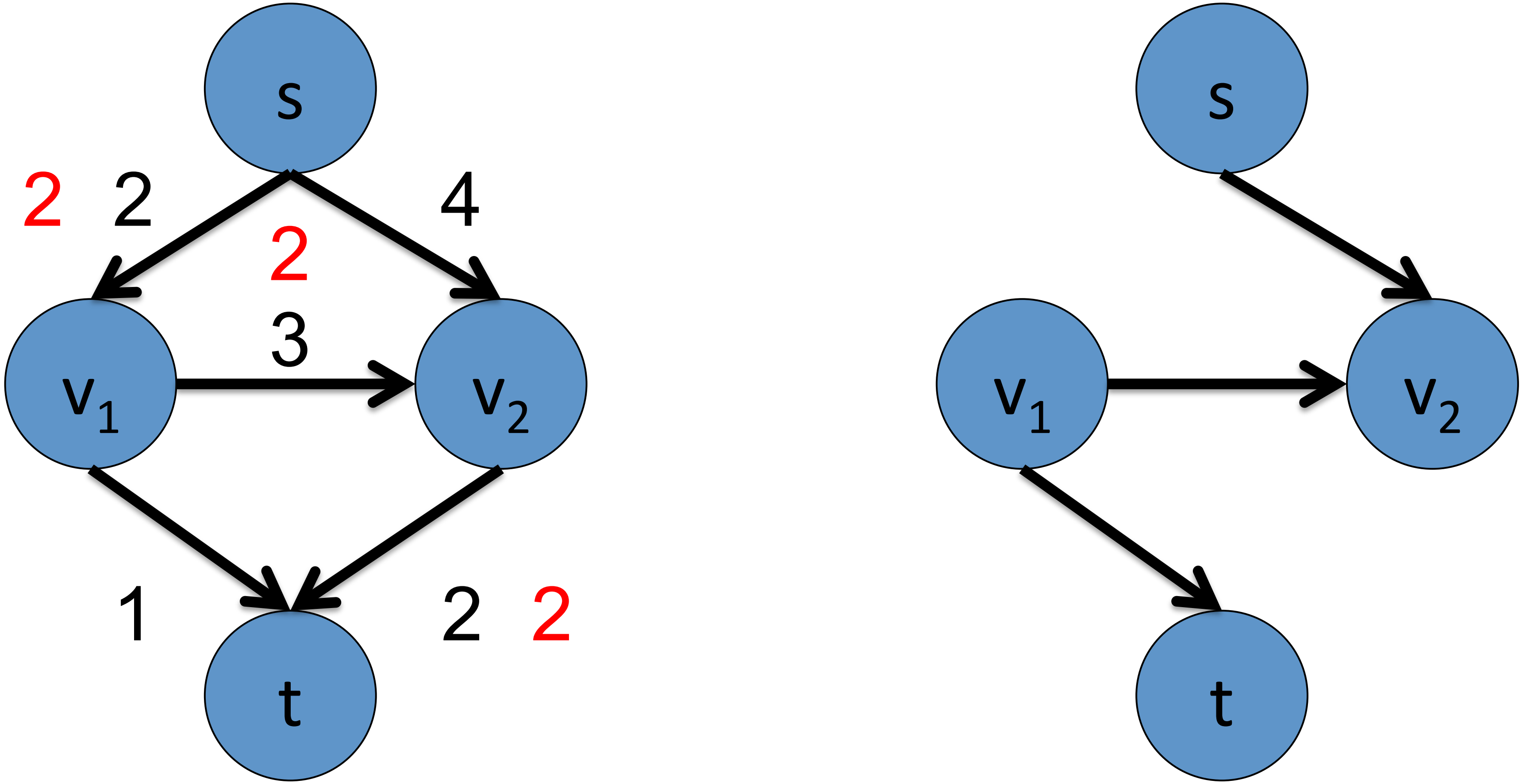
Outline

- Preliminaries

- Maximum Flow
  - **Residual Graph**
  - Max-Flow Min-Cut Theorem

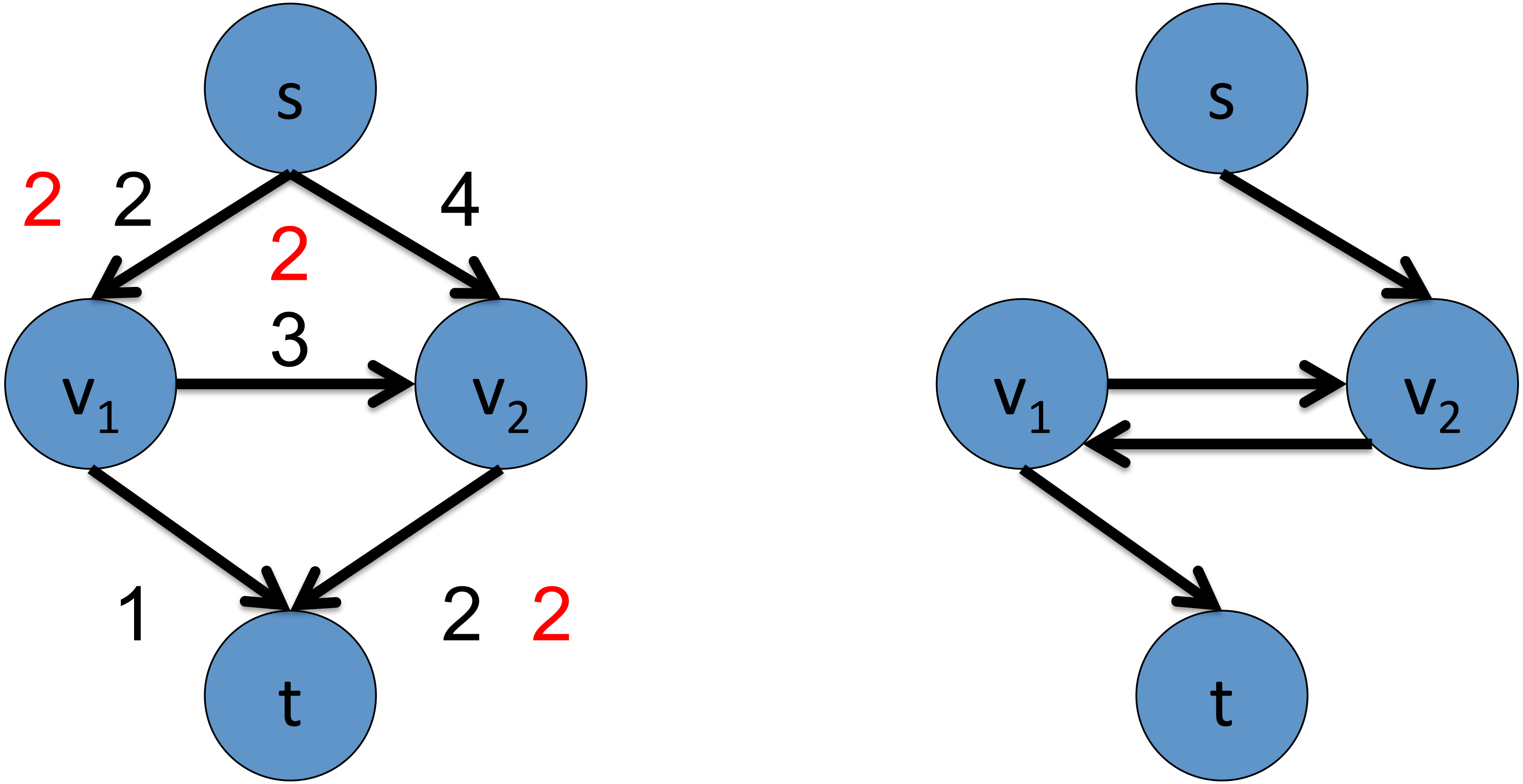
- Algorithms

Residual Graph



Arcs where flow(a) < c(a)

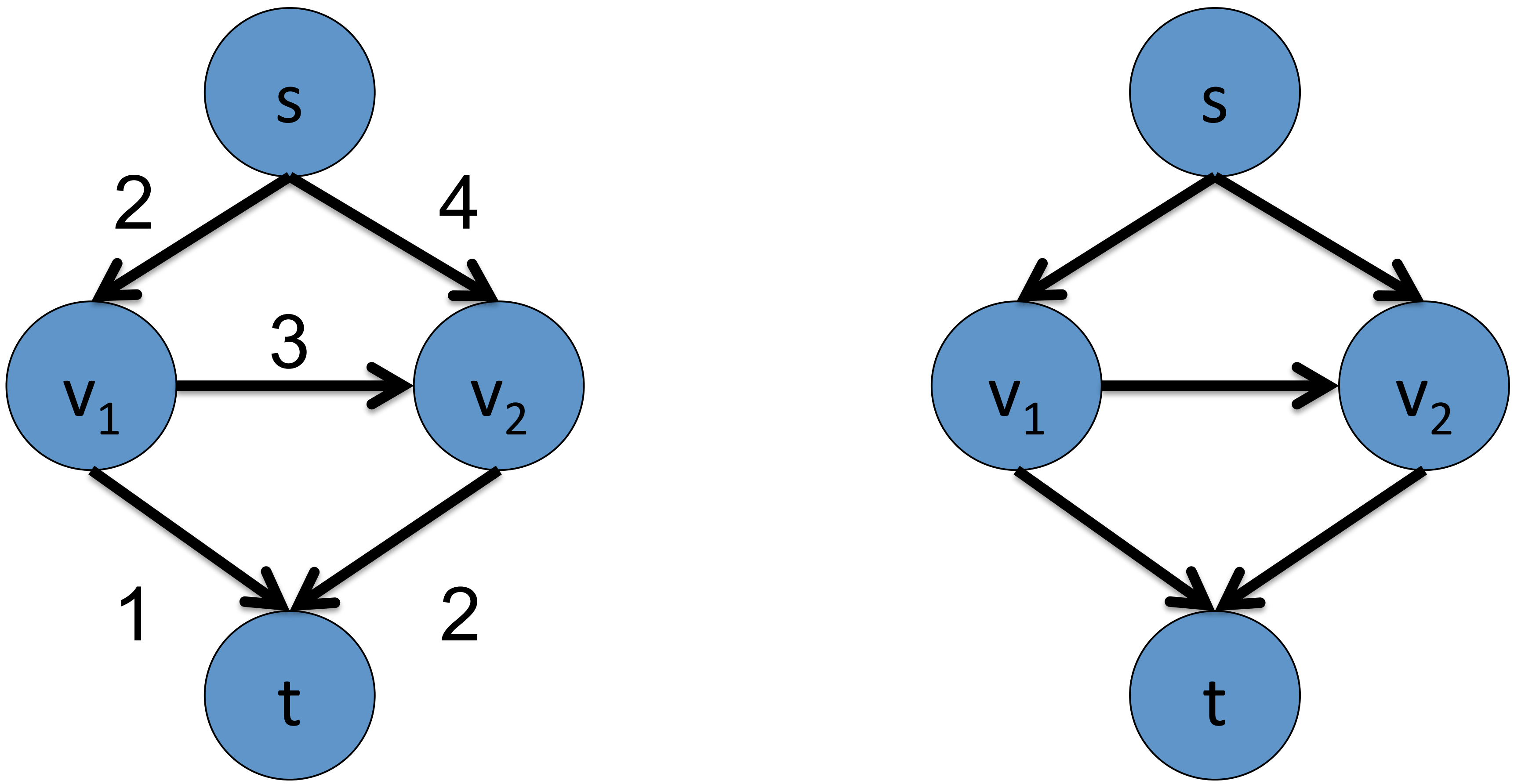
Residual Graph



Including arcs to s and from t is not necessary

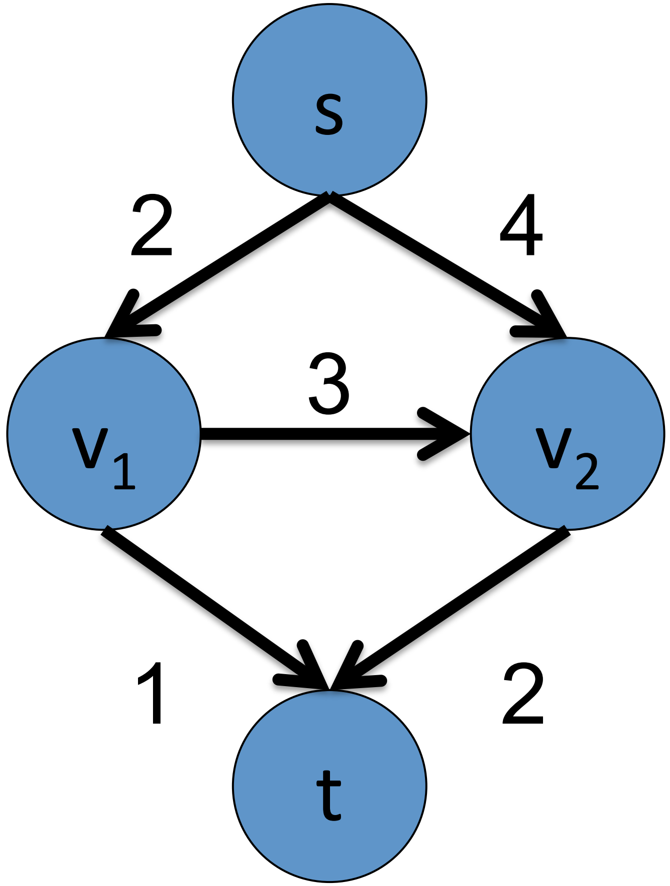
Inverse of arcs where flow(a) > 0

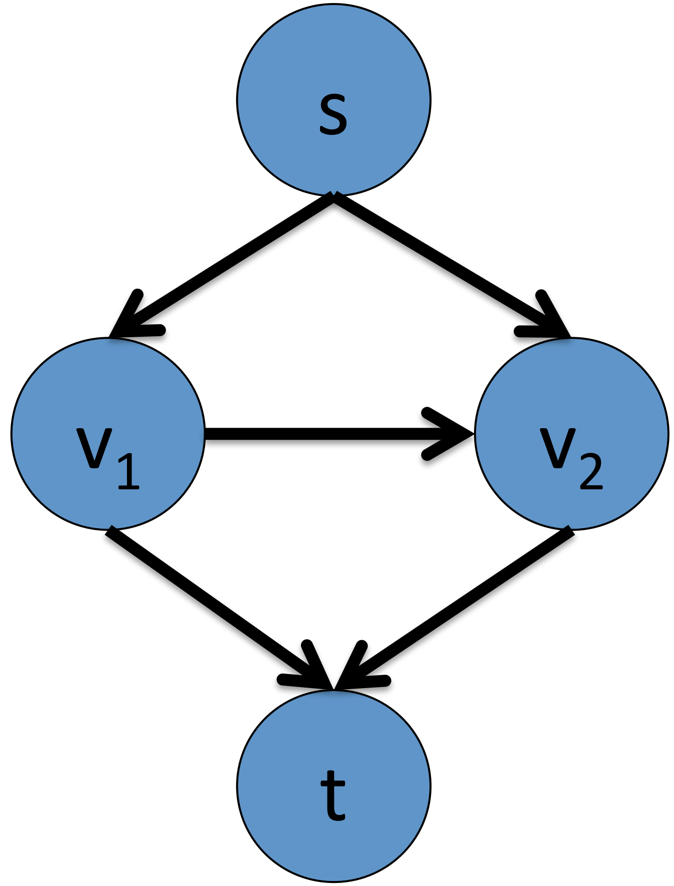
Maximum Flow using Residual Graphs



Start with zero flow.

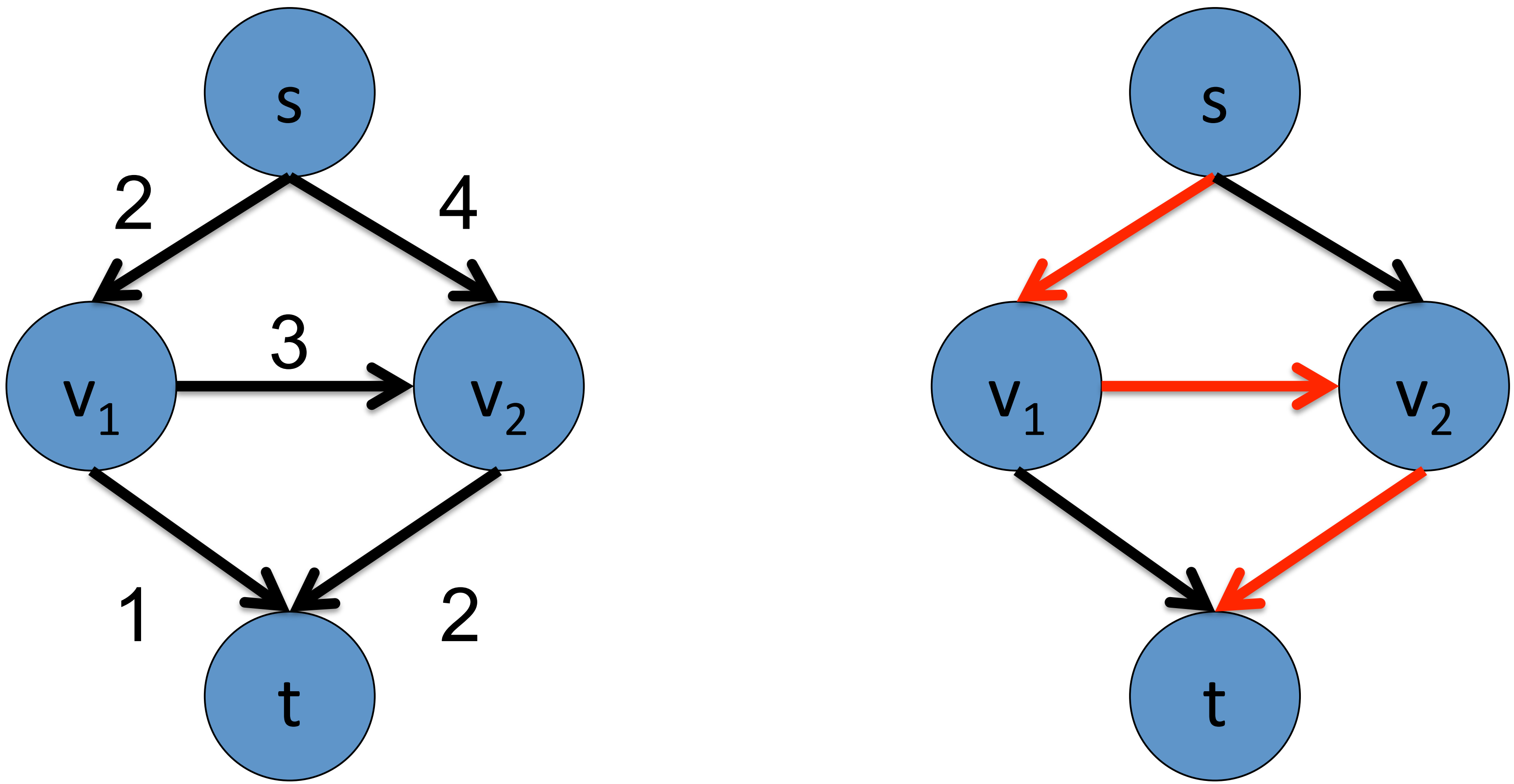
Maximum Flow using Residual Graphs





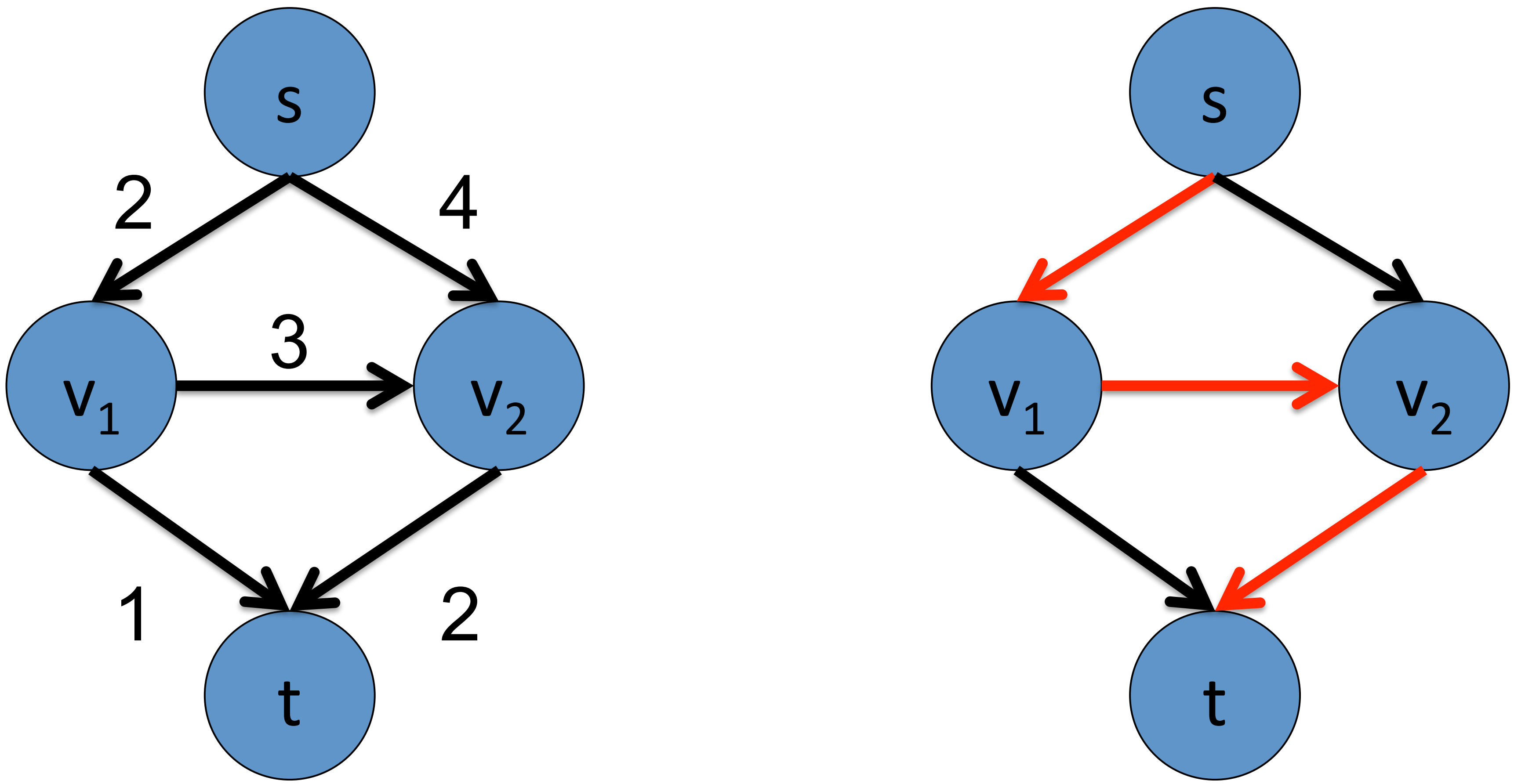
Find an s-t path in the residual graph.

Maximum Flow using Residual Graphs



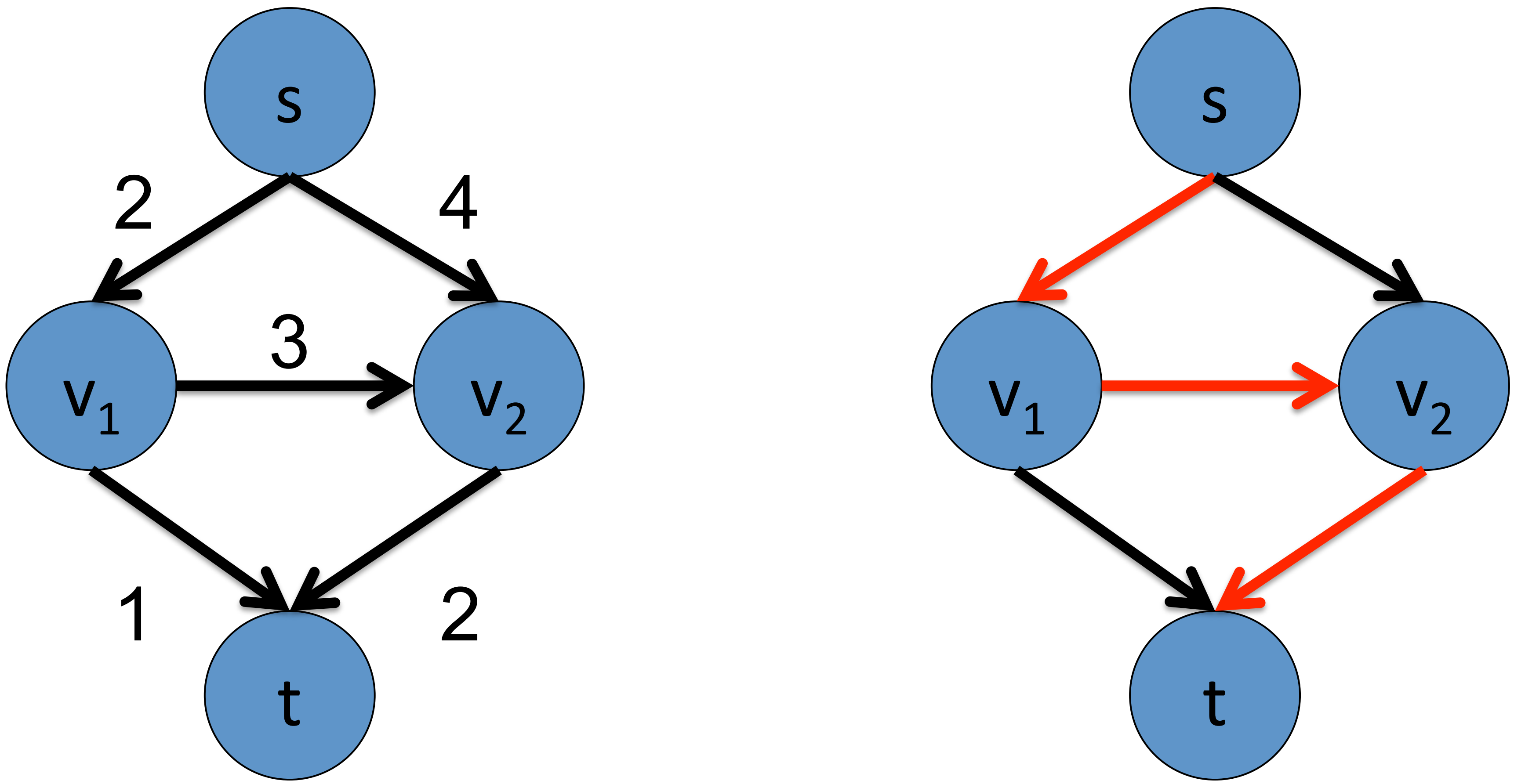
Find an s-t path in the residual graph.

Maximum Flow using Residual Graphs



For inverse arcs in path, subtract flow K.

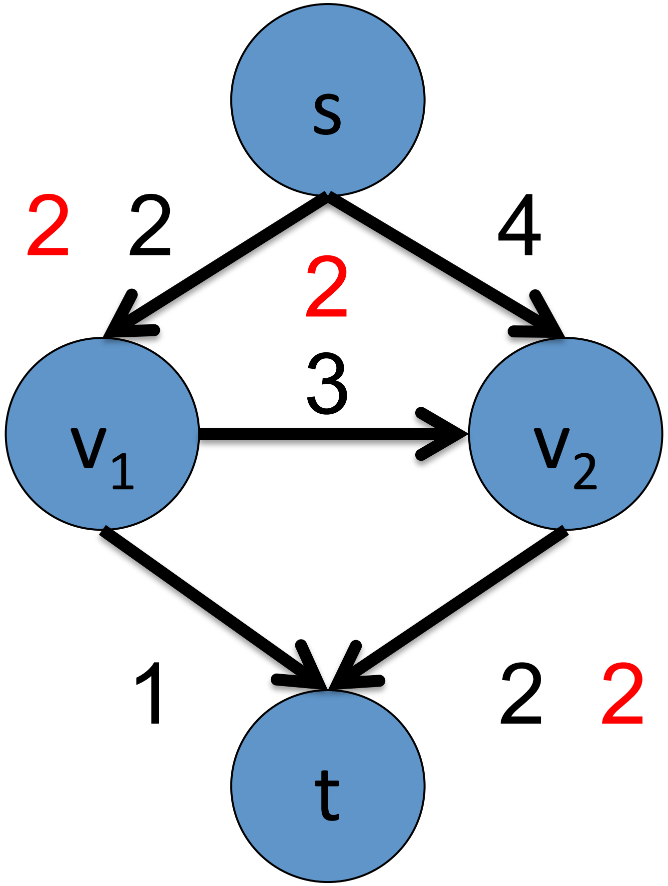
Maximum Flow using Residual Graphs

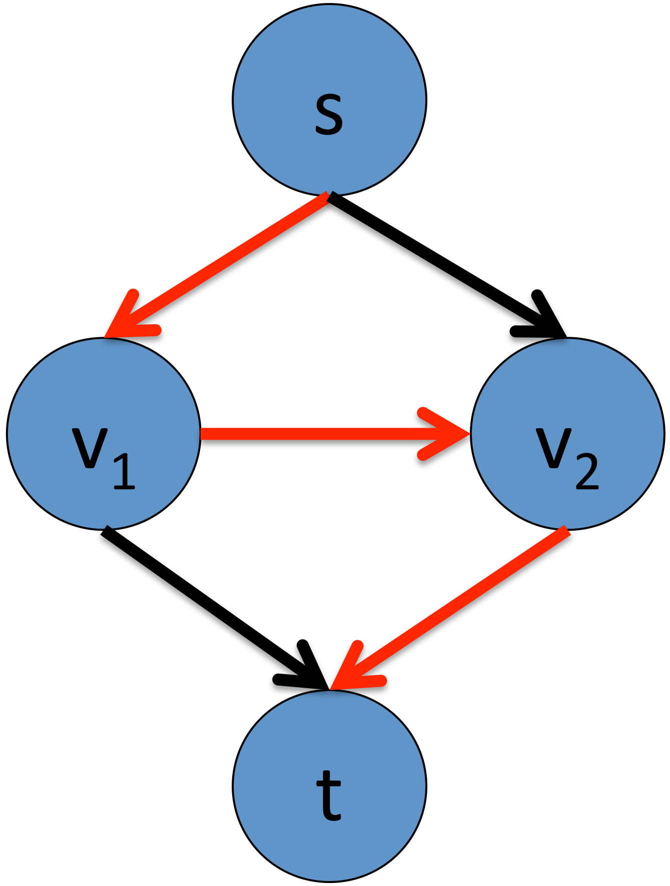


Choose maximum allowable value of K.

For forward arcs in path, add flow K.

Maximum Flow using Residual Graphs

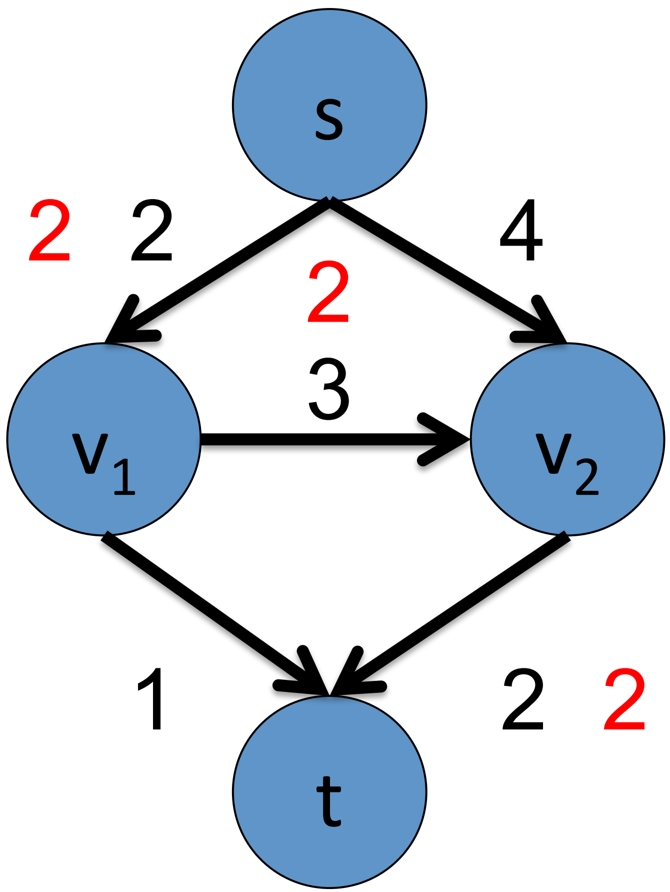


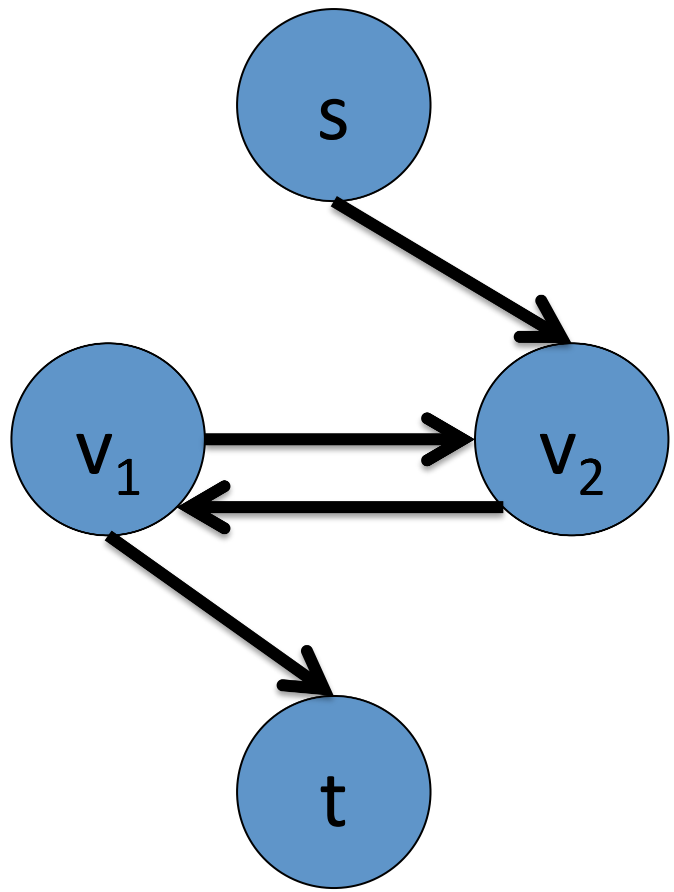


Choose maximum allowable value of K.

For forward arcs in path, add flow K.

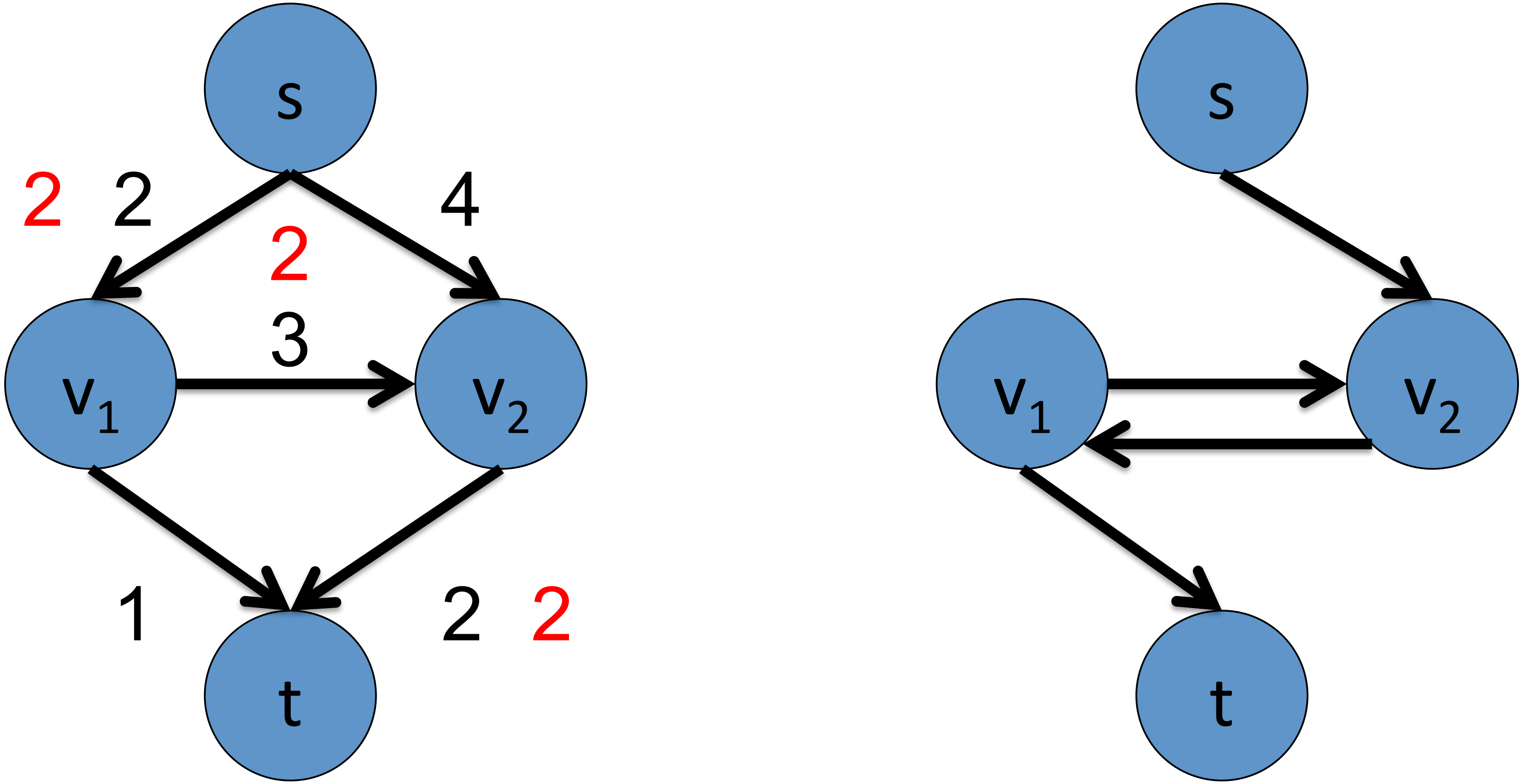
Maximum Flow using Residual Graphs





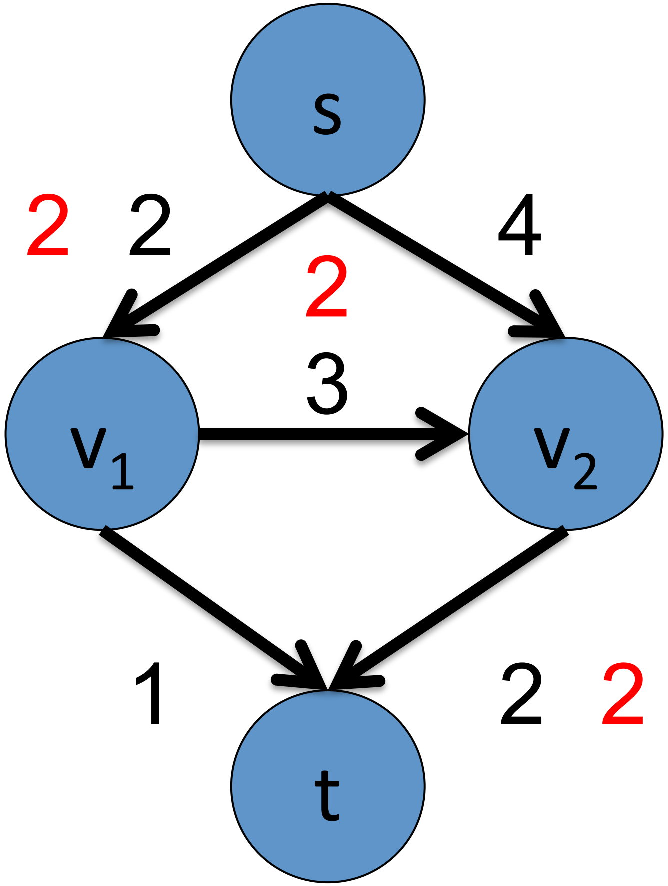
Update the residual graph.

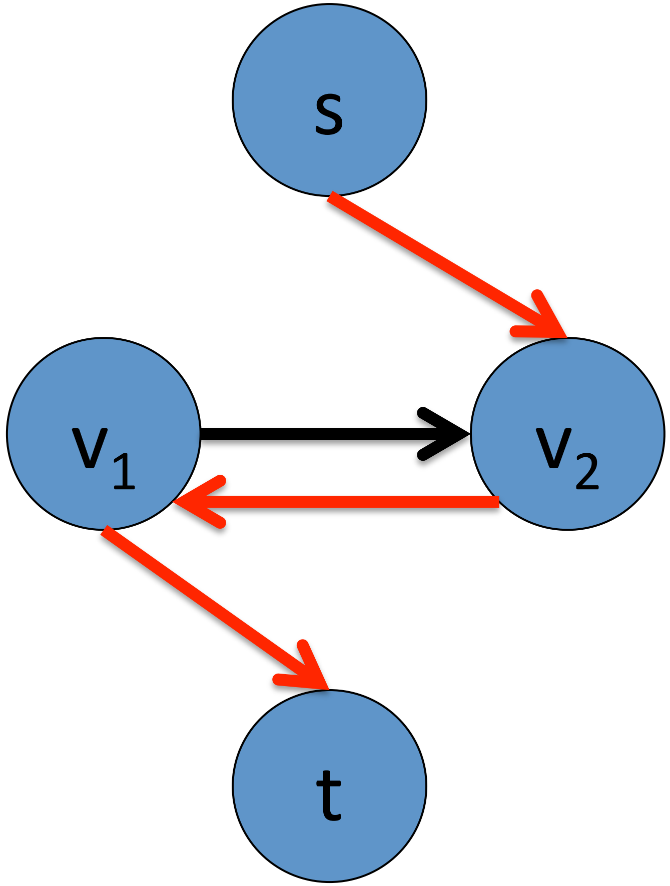
Maximum Flow using Residual Graphs



Find an s-t path in the residual graph.

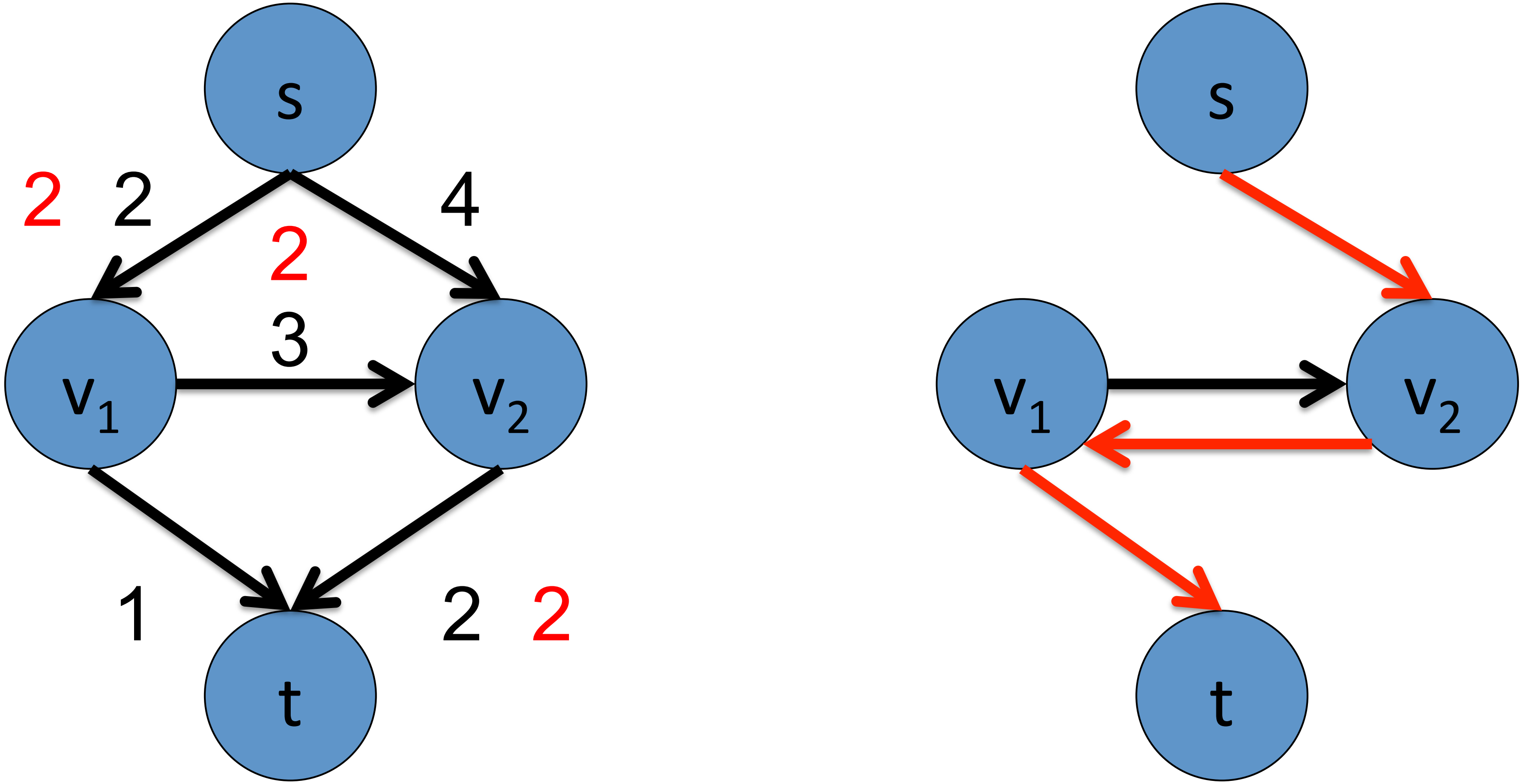
Maximum Flow using Residual Graphs





Find an s-t path in the residual graph.

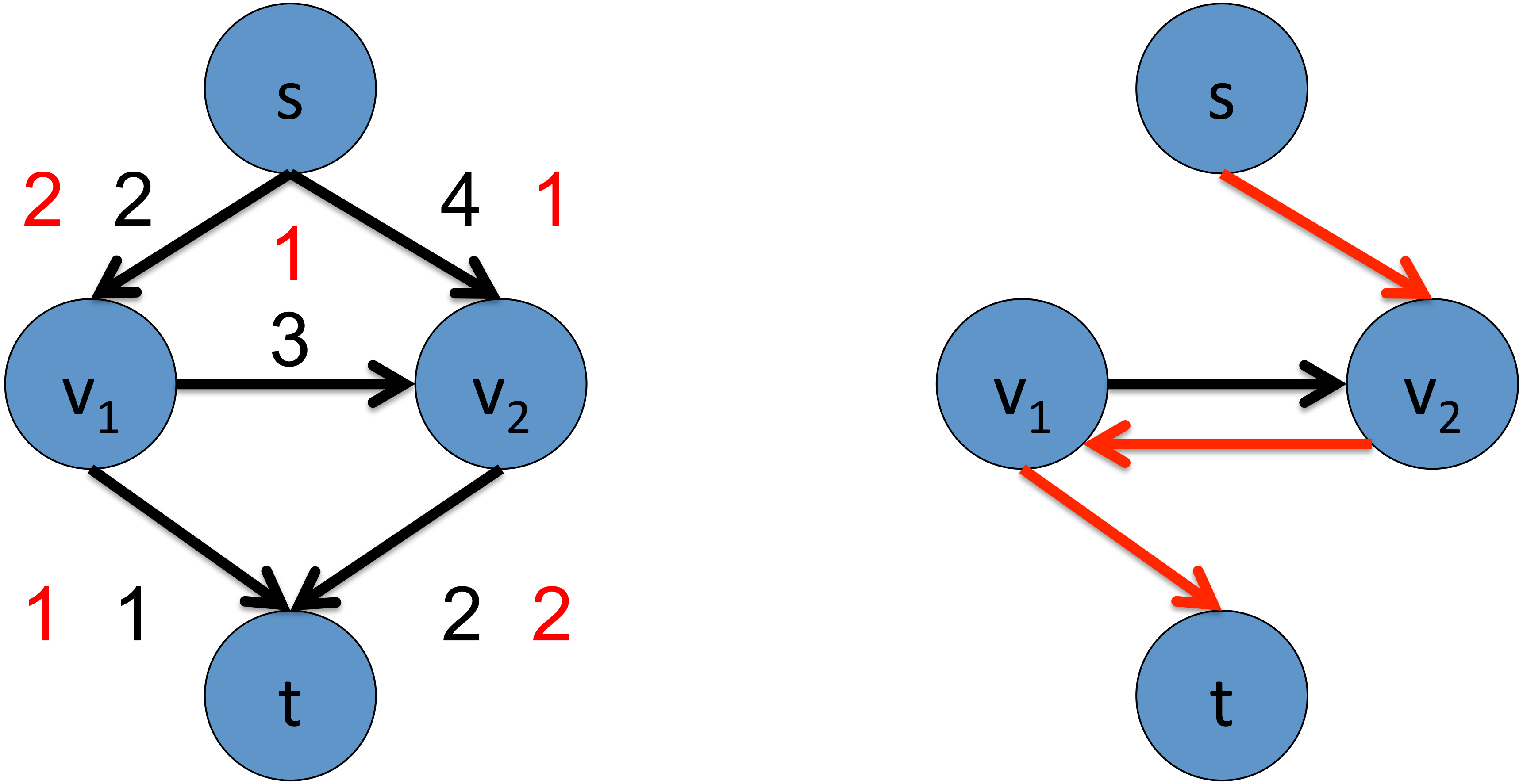
Maximum Flow using Residual Graphs



Choose maximum allowable value of K.

Add K to (s,v_2) and (v_1,t). Subtract K from (v_1,v_2).

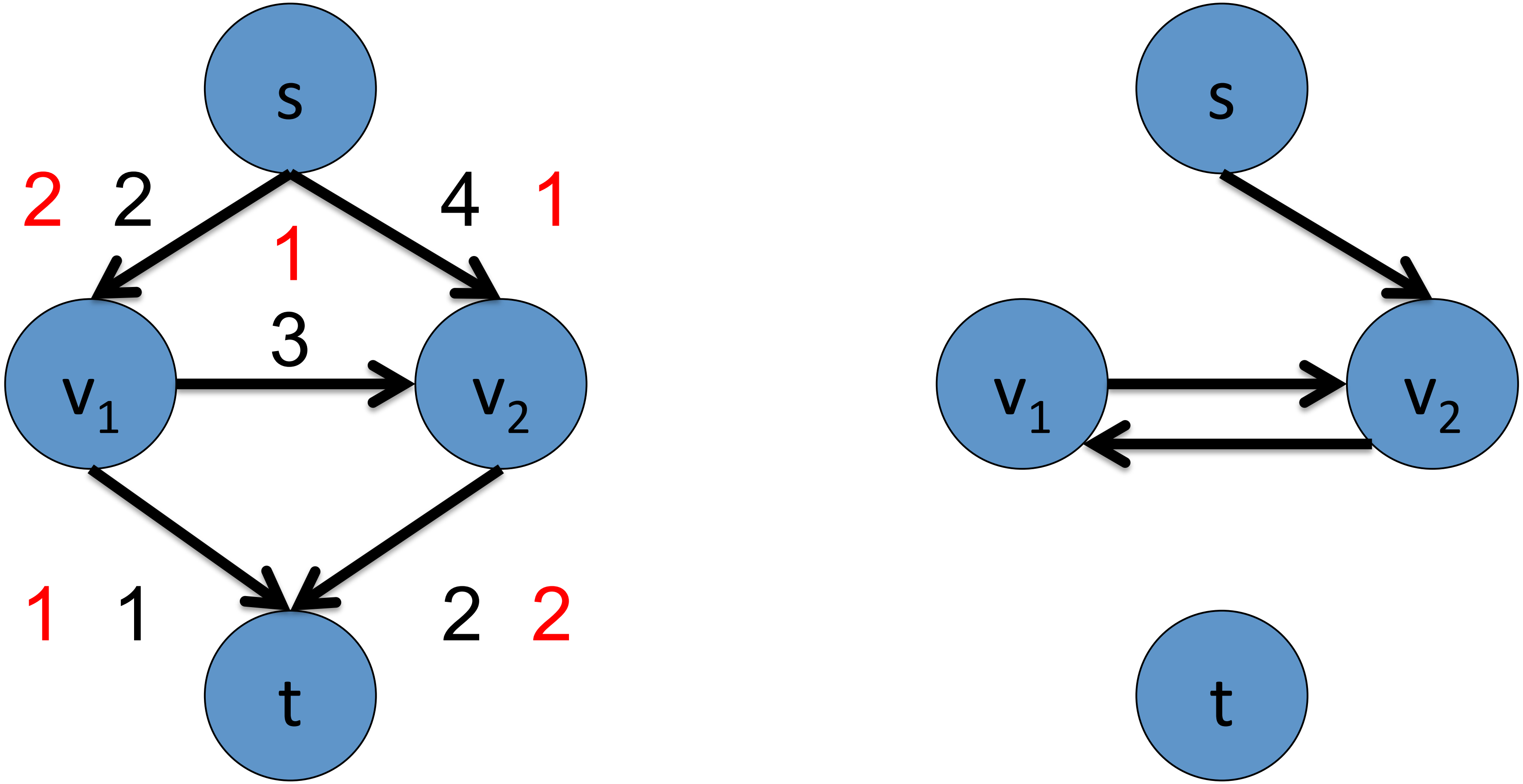
Maximum Flow using Residual Graphs



Choose maximum allowable value of K.

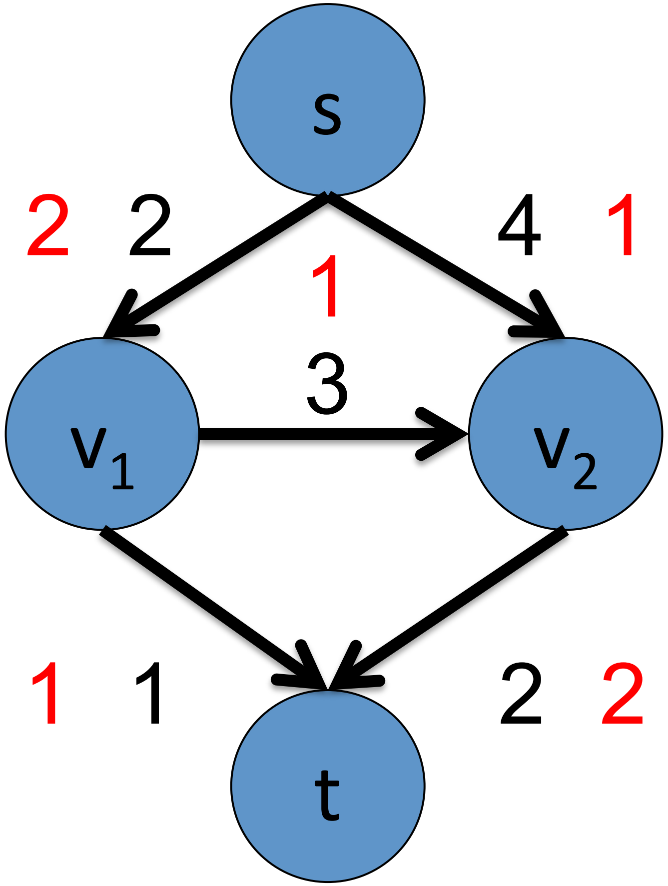
Add K to (s,v_2) and (v_1,t). Subtract K from (v_1,v_2).

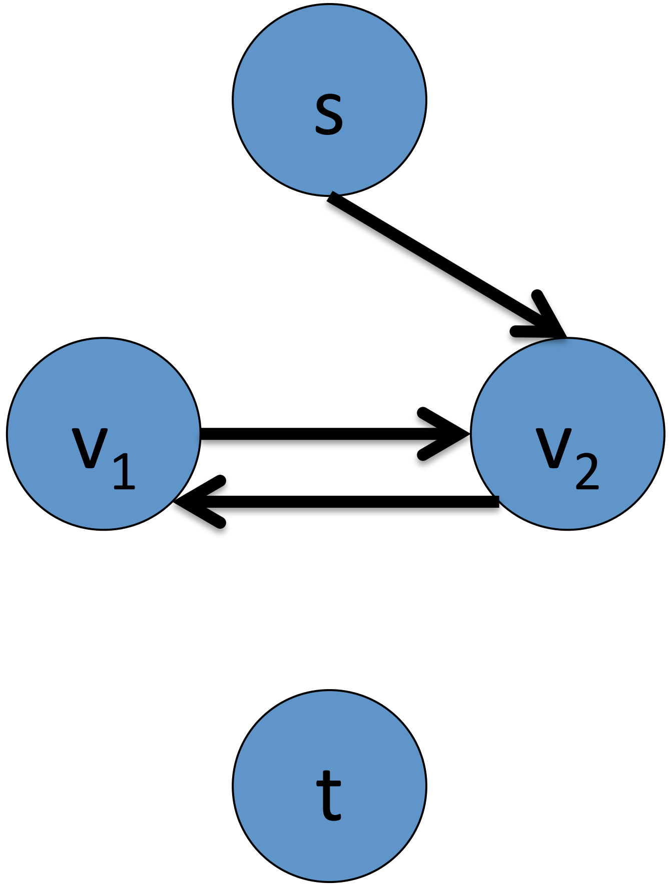
Maximum Flow using Residual Graphs



Update the residual graph.

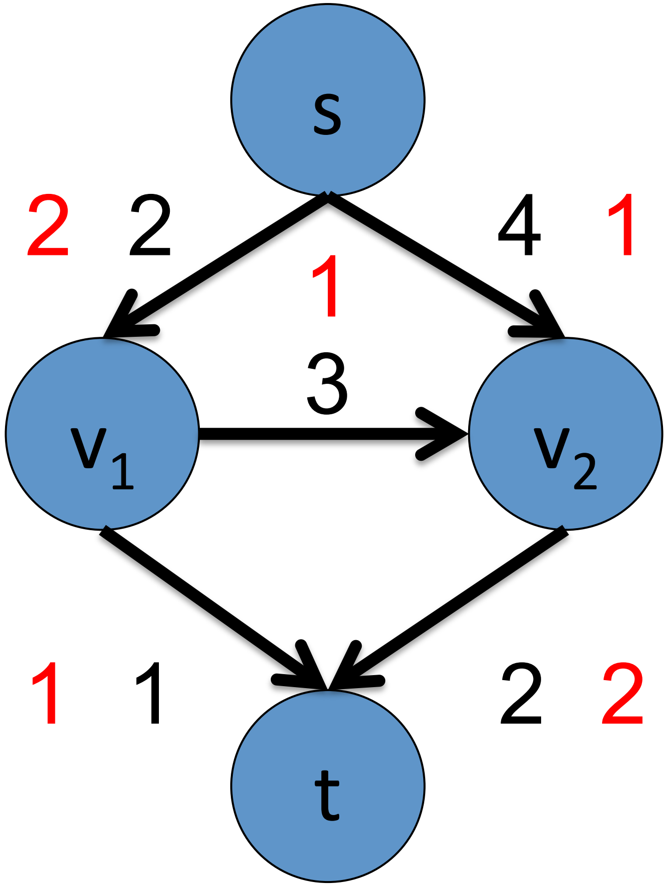
Maximum Flow using Residual Graphs

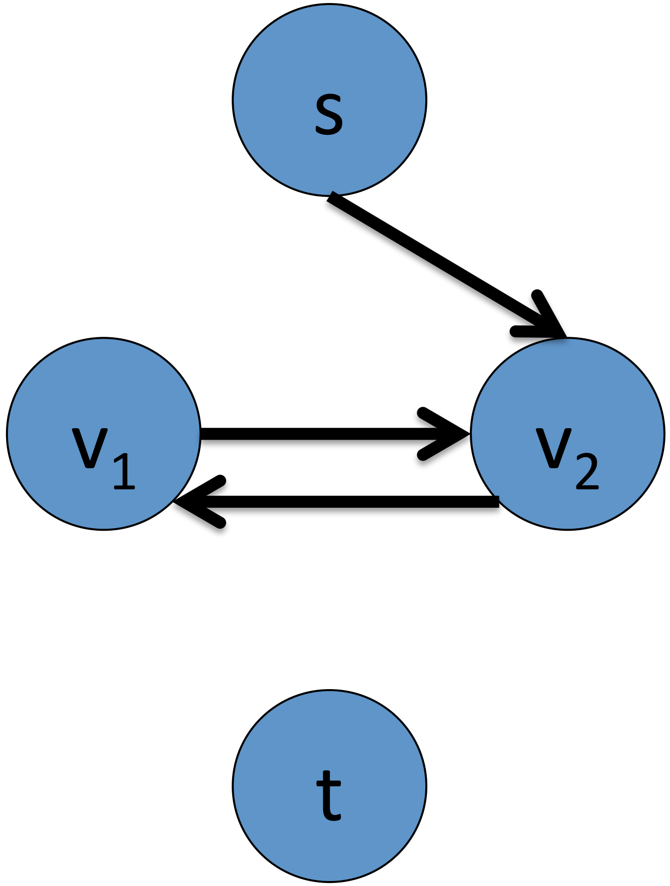




Find an s-t path in the residual graph.

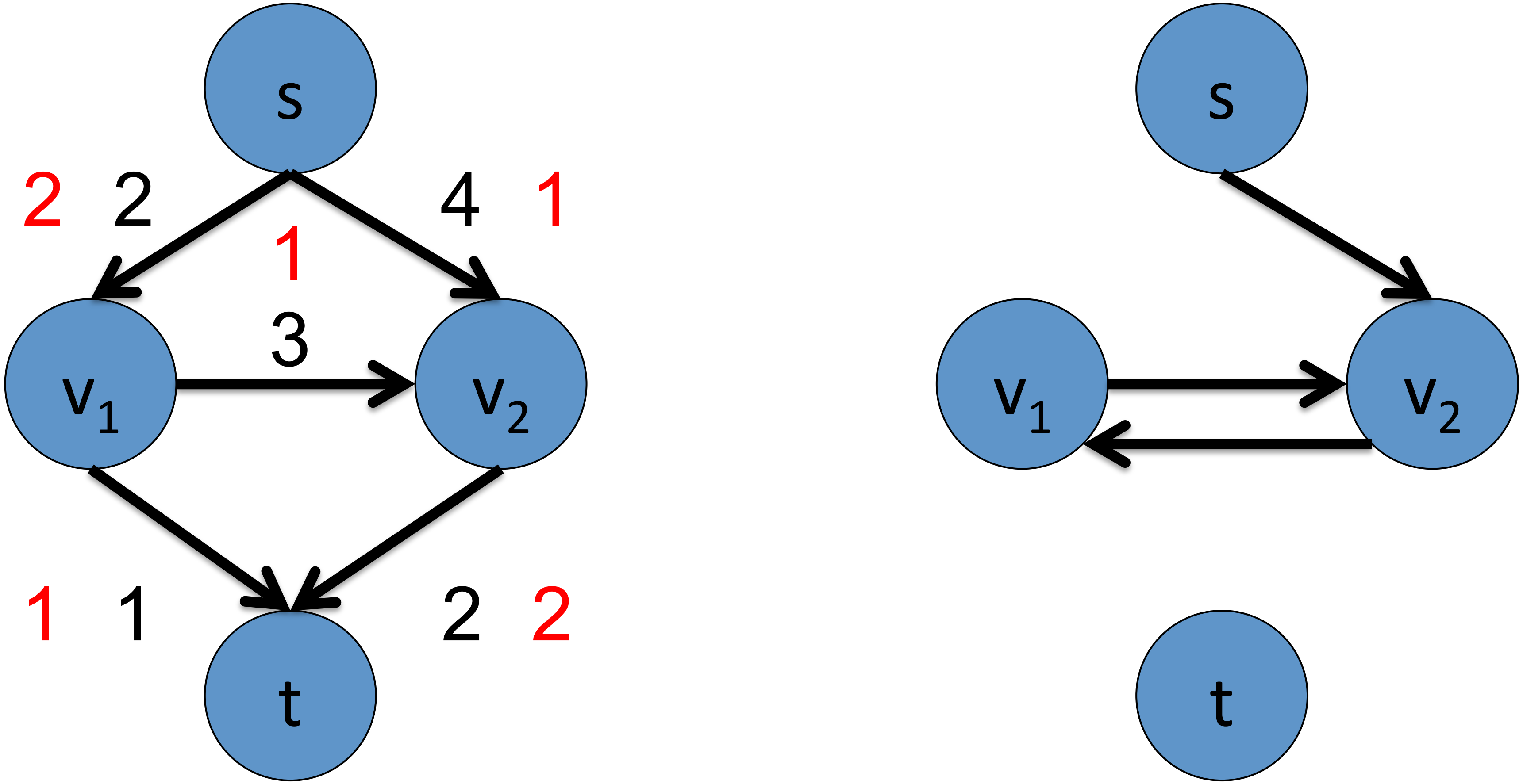
Maximum Flow using Residual Graphs





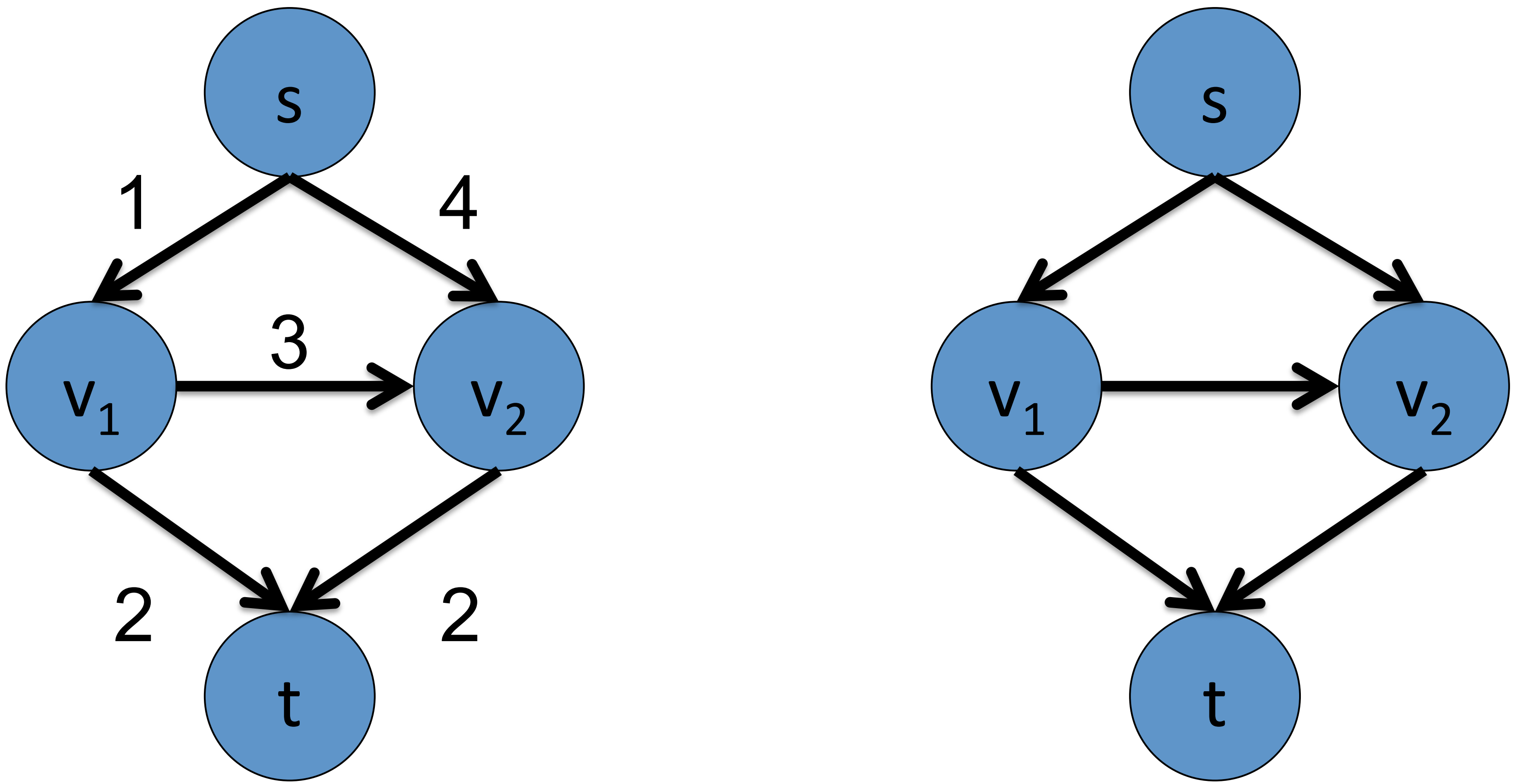
No more s-t paths. Stop.

Maximum Flow using Residual Graphs



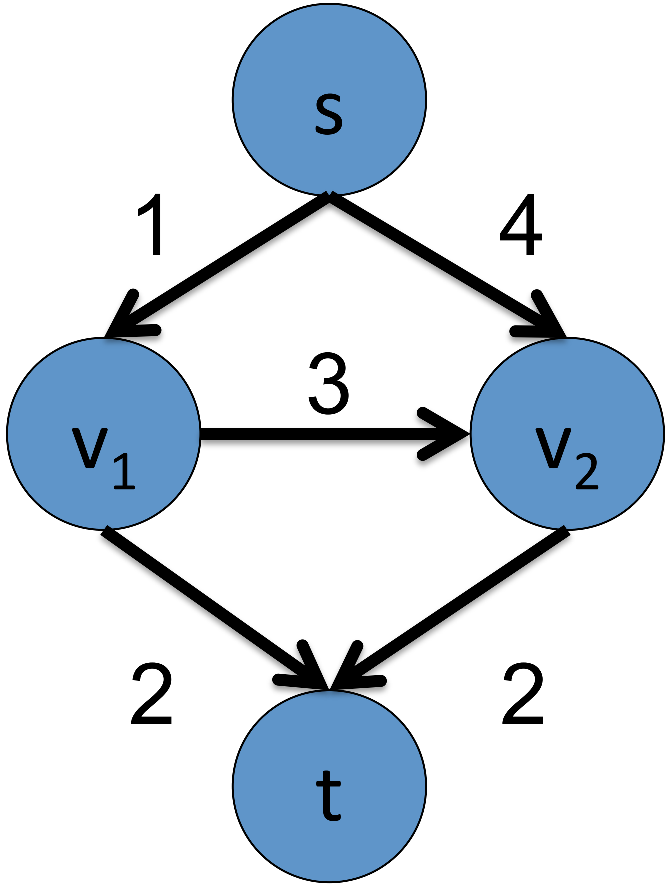
Correct Answer.

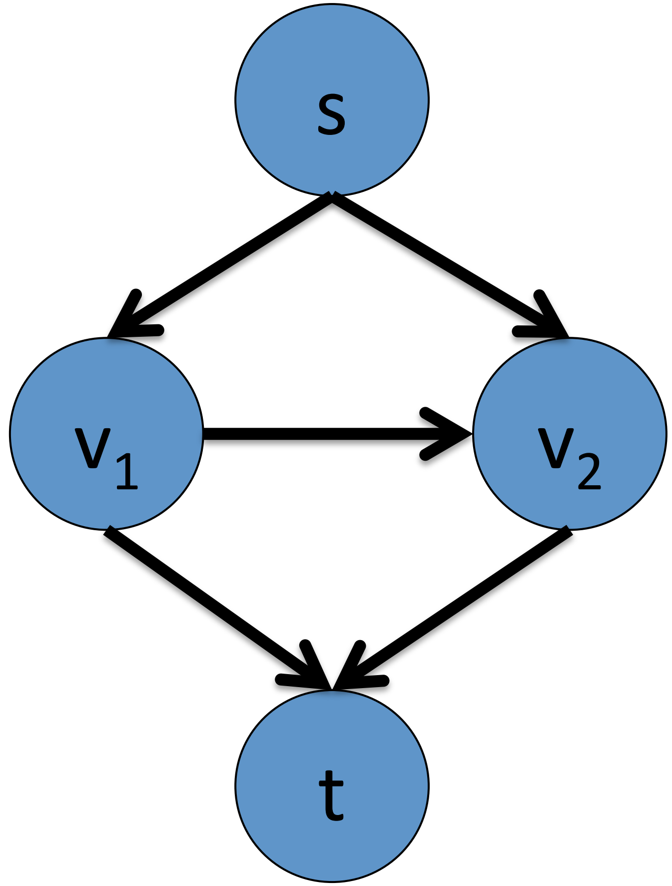
Maximum Flow using Residual Graphs



Start with zero flow

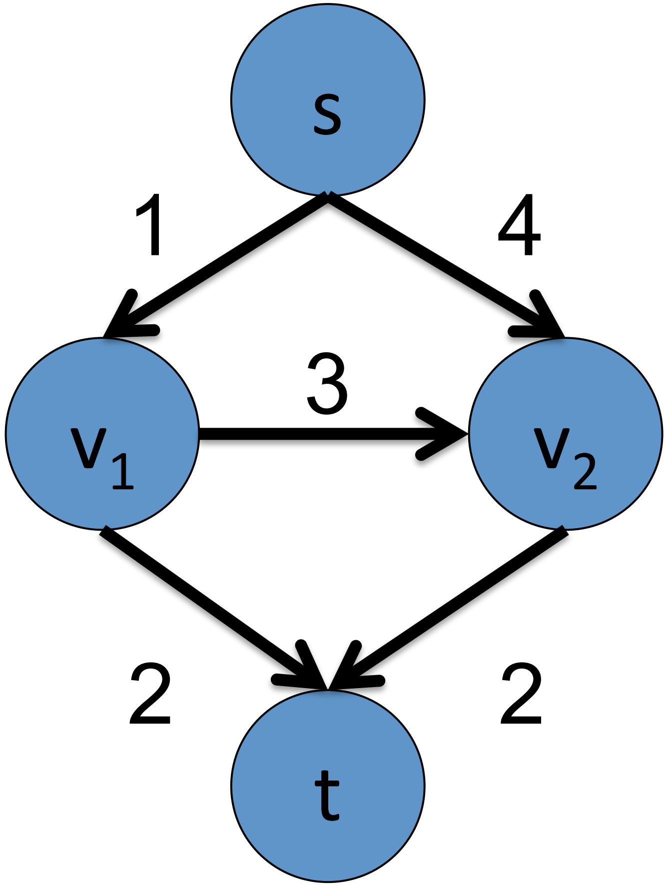
Maximum Flow using Residual Graphs

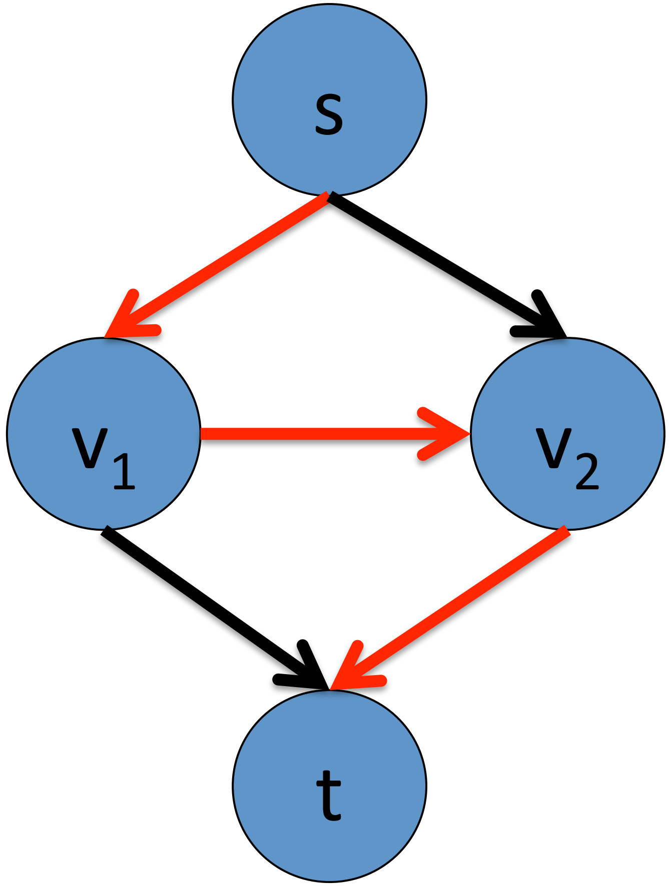




Find an s-t path in the residual graph.

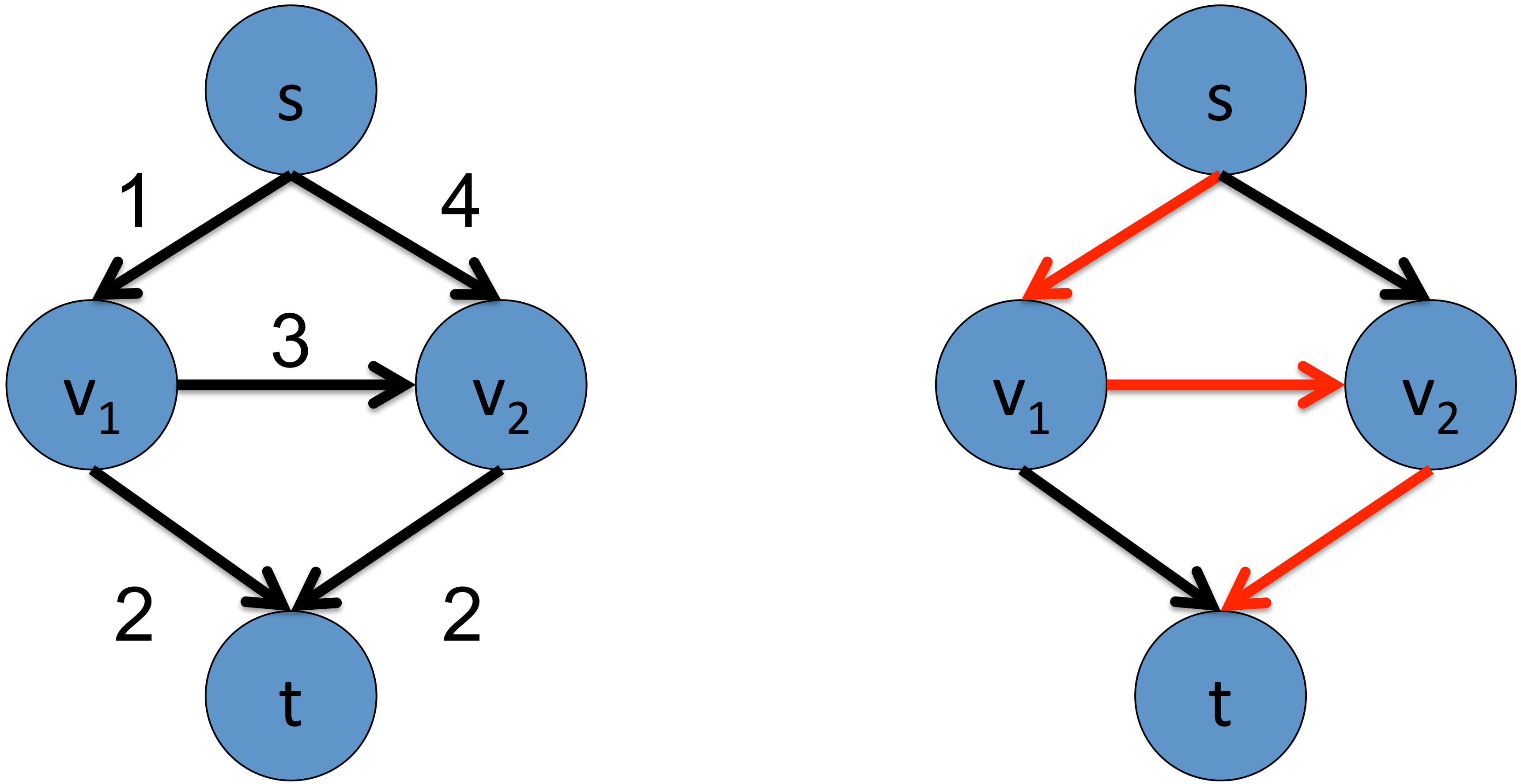
Maximum Flow using Residual Graphs





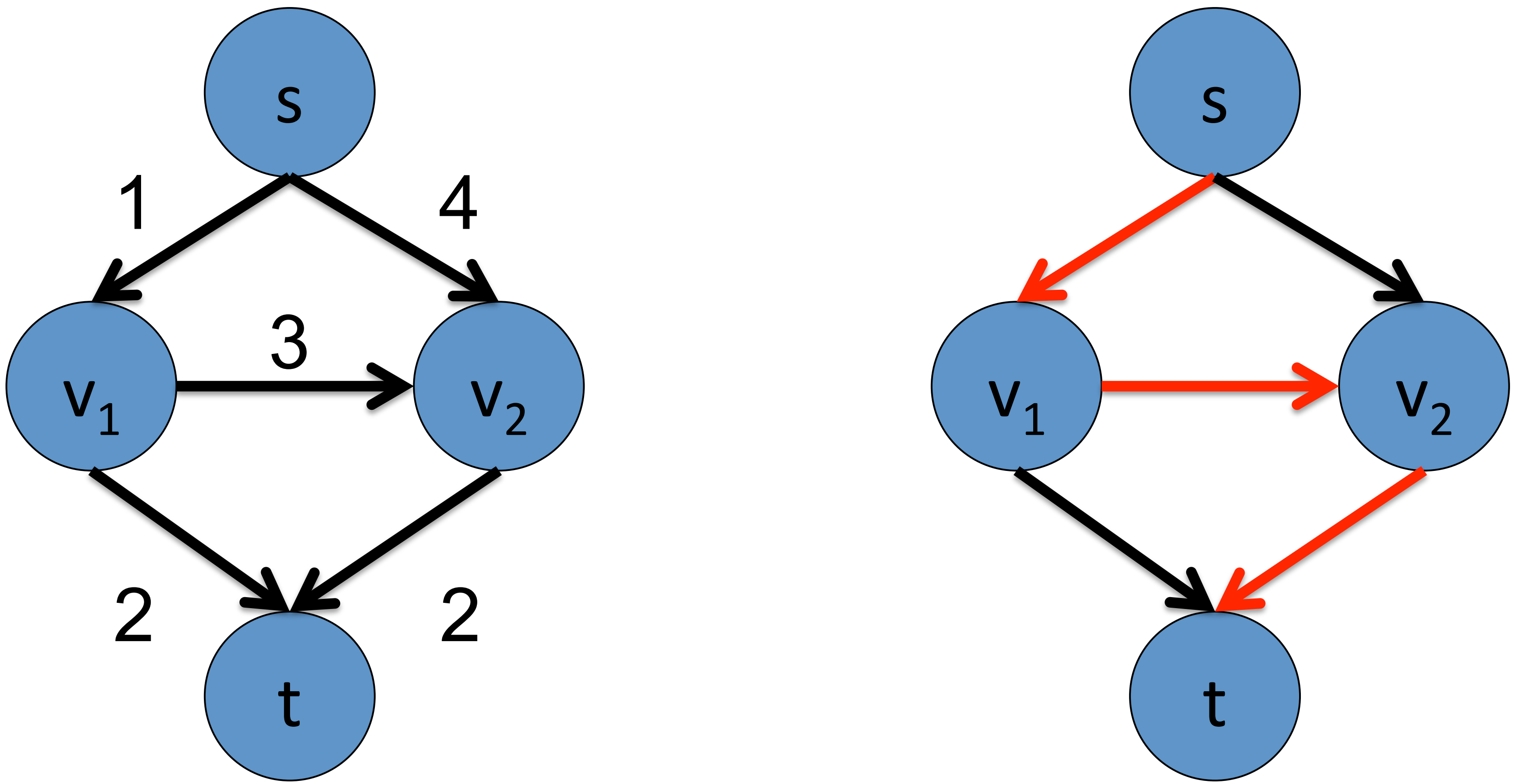
Find an s-t path in the residual graph.

Maximum Flow using Residual Graphs



For inverse arcs in path, subtract flow K.

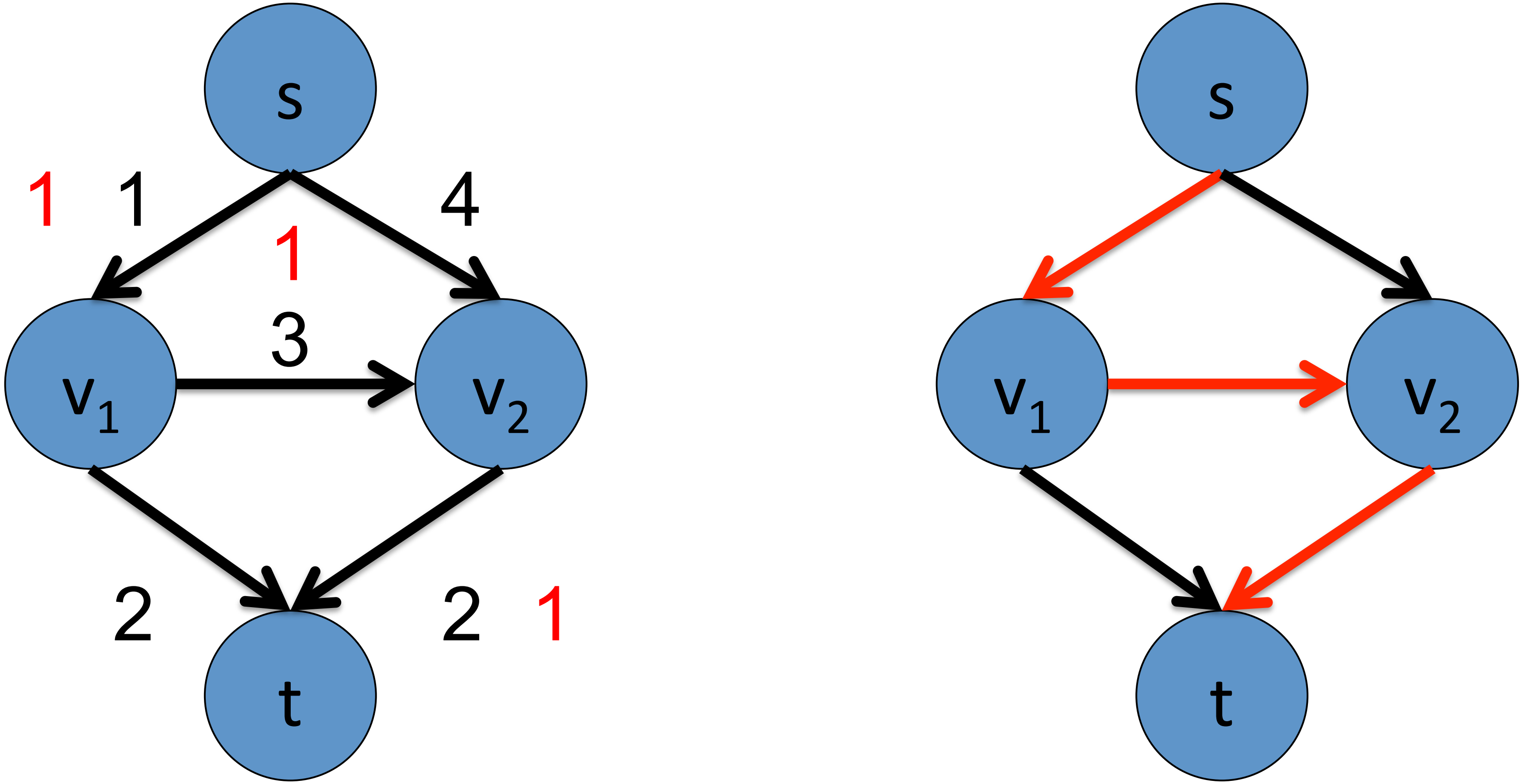
Maximum Flow using Residual Graphs



Choose maximum allowable value of K.

For forward arcs in path, add flow K.

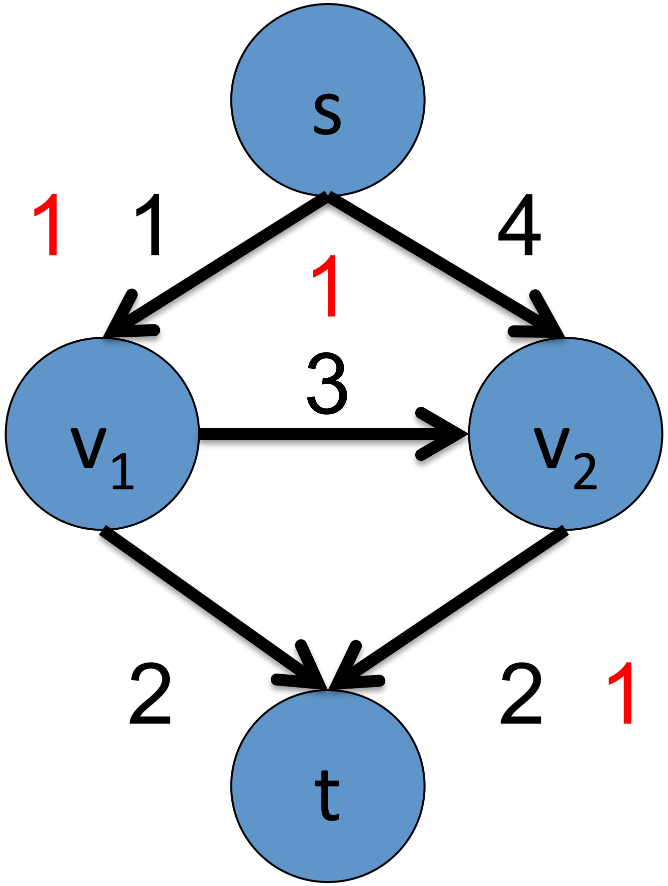
Maximum Flow using Residual Graphs

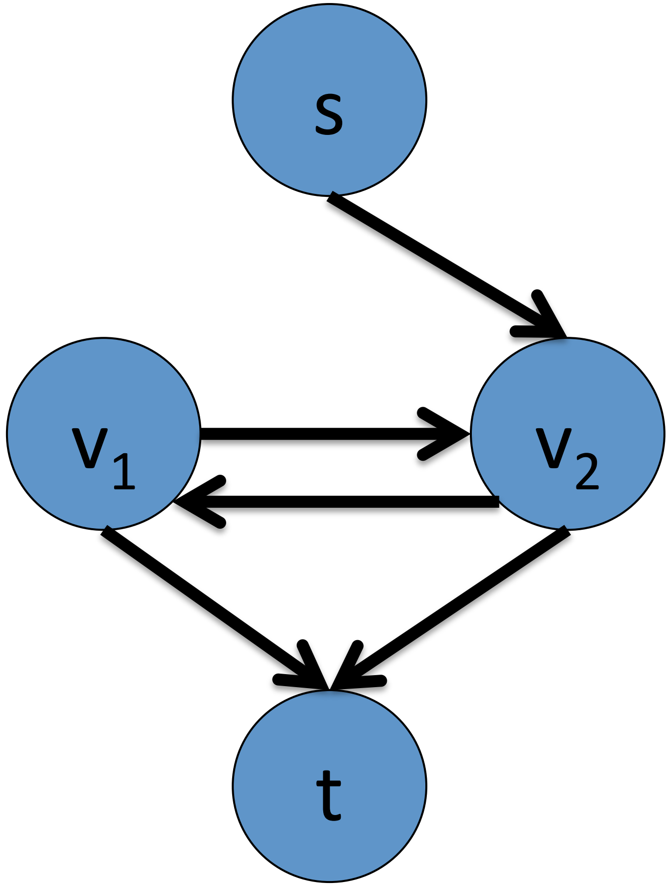


Choose maximum allowable value of K.

For forward arcs in path, add flow K.

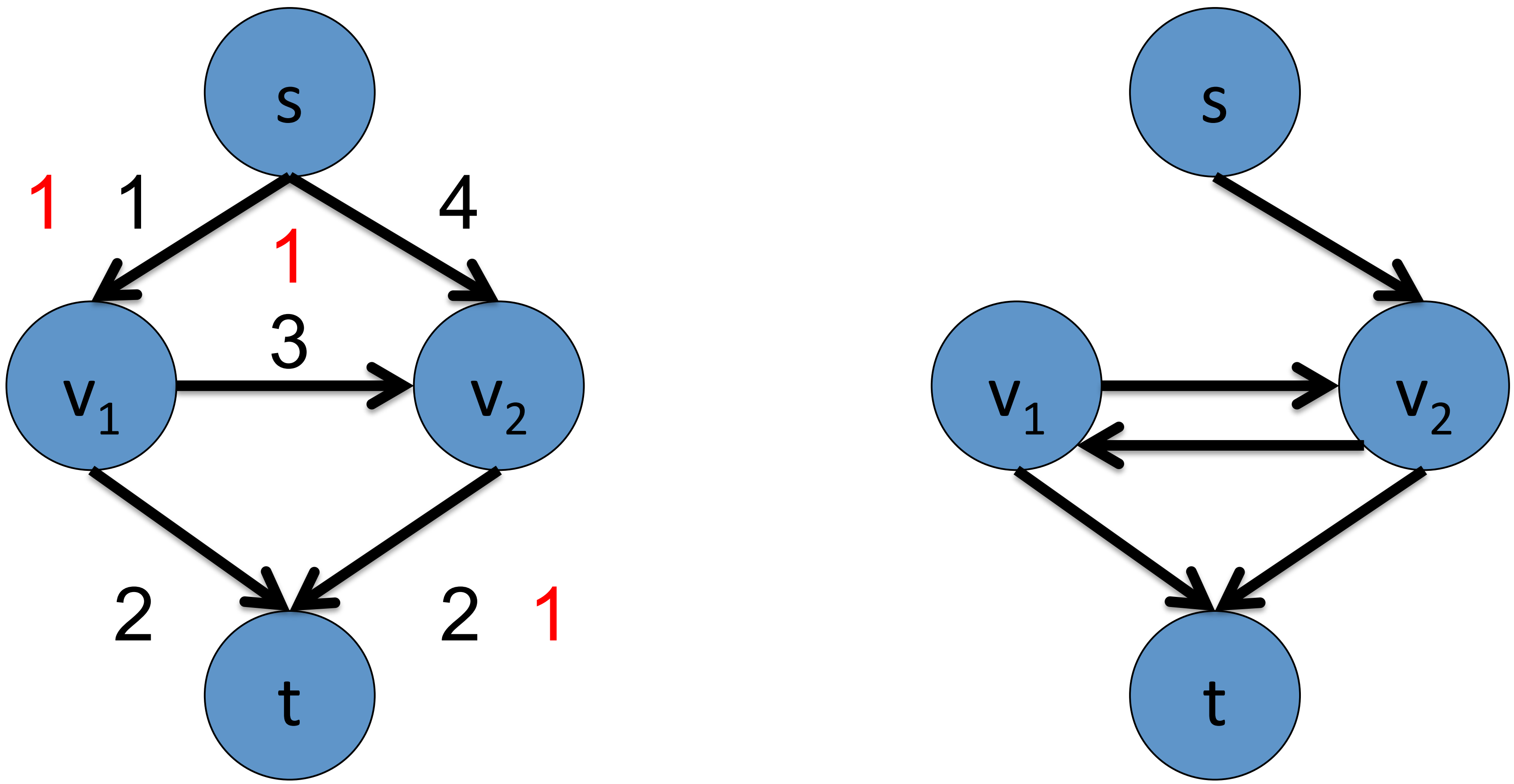
Maximum Flow using Residual Graphs





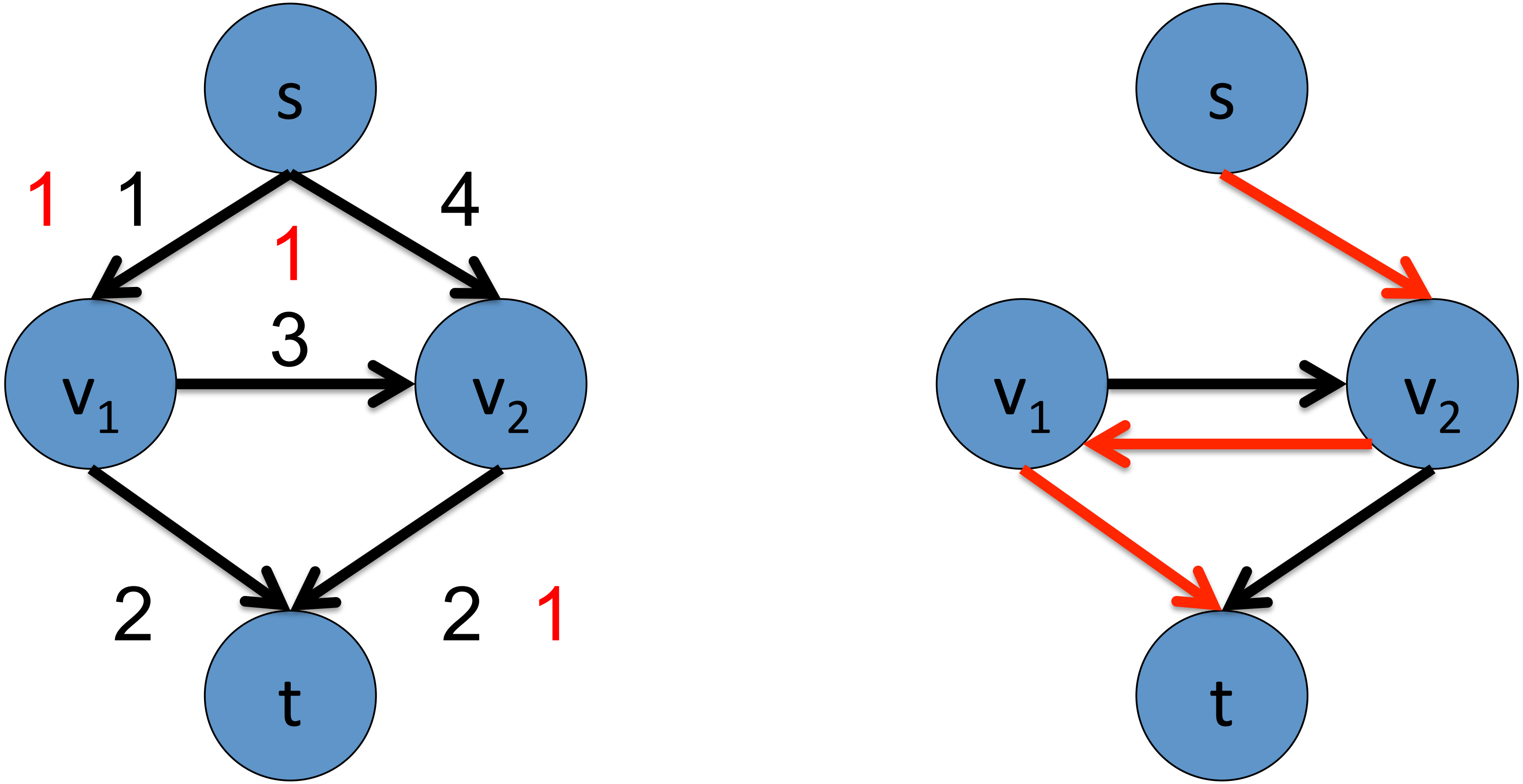
Update the residual graph.

Maximum Flow using Residual Graphs



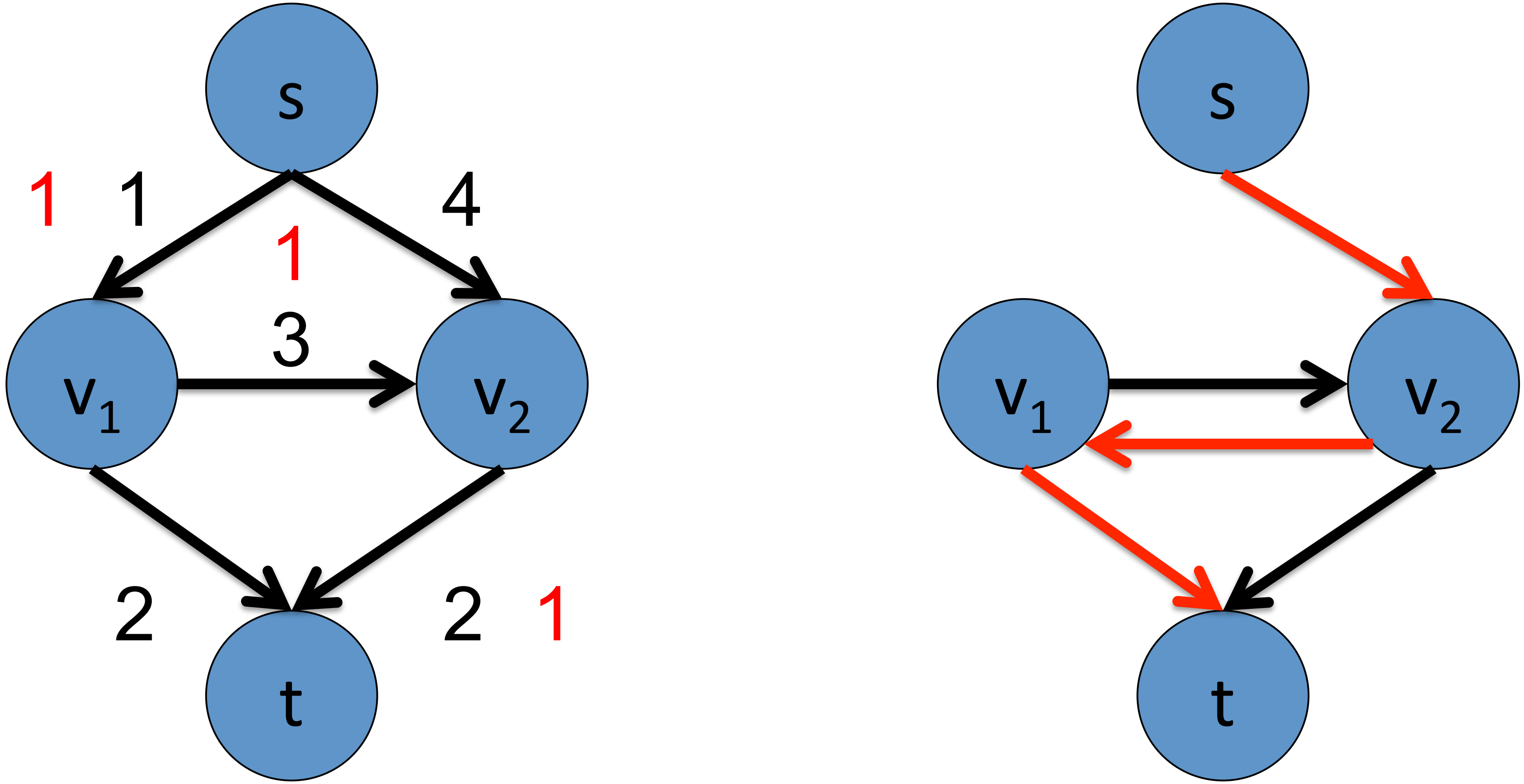
Find an s-t path in the residual graph.

Maximum Flow using Residual Graphs



Find an s-t path in the residual graph.

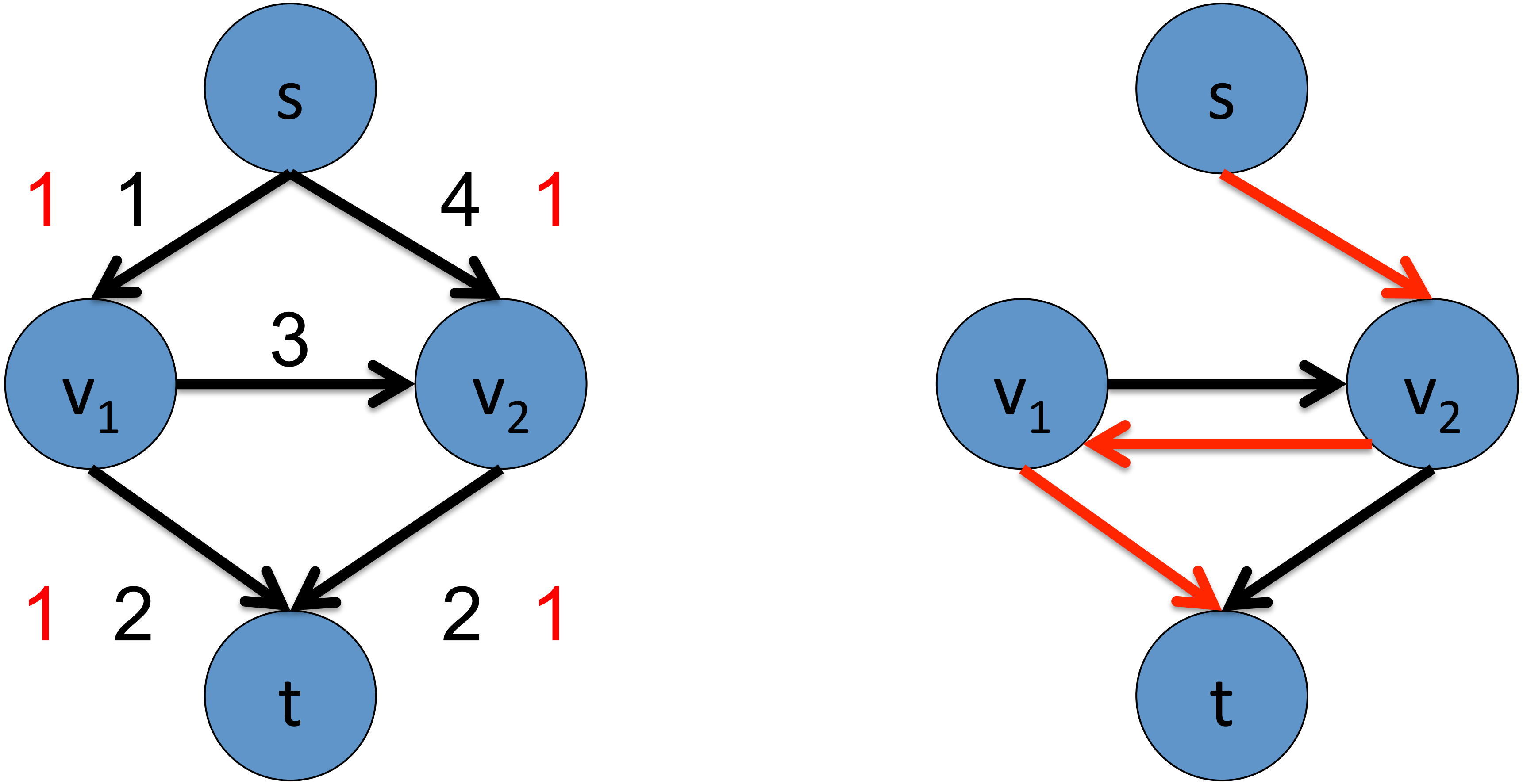
Maximum Flow using Residual Graphs



Choose maximum allowable value of K.

Add K to (s,v_2) and (v_1,t). Subtract K from (v_1,v_2).

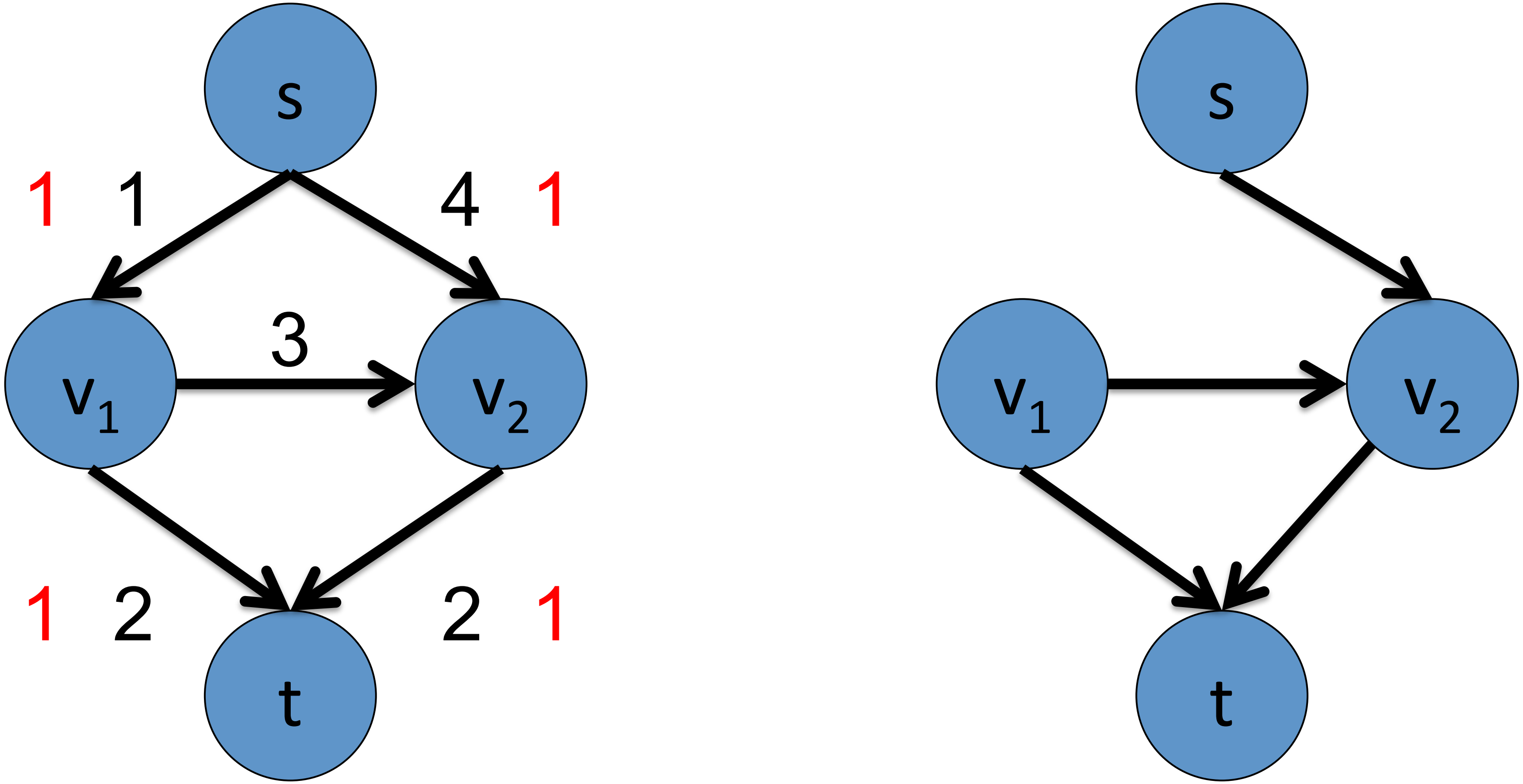
Maximum Flow using Residual Graphs



Choose maximum allowable value of K.

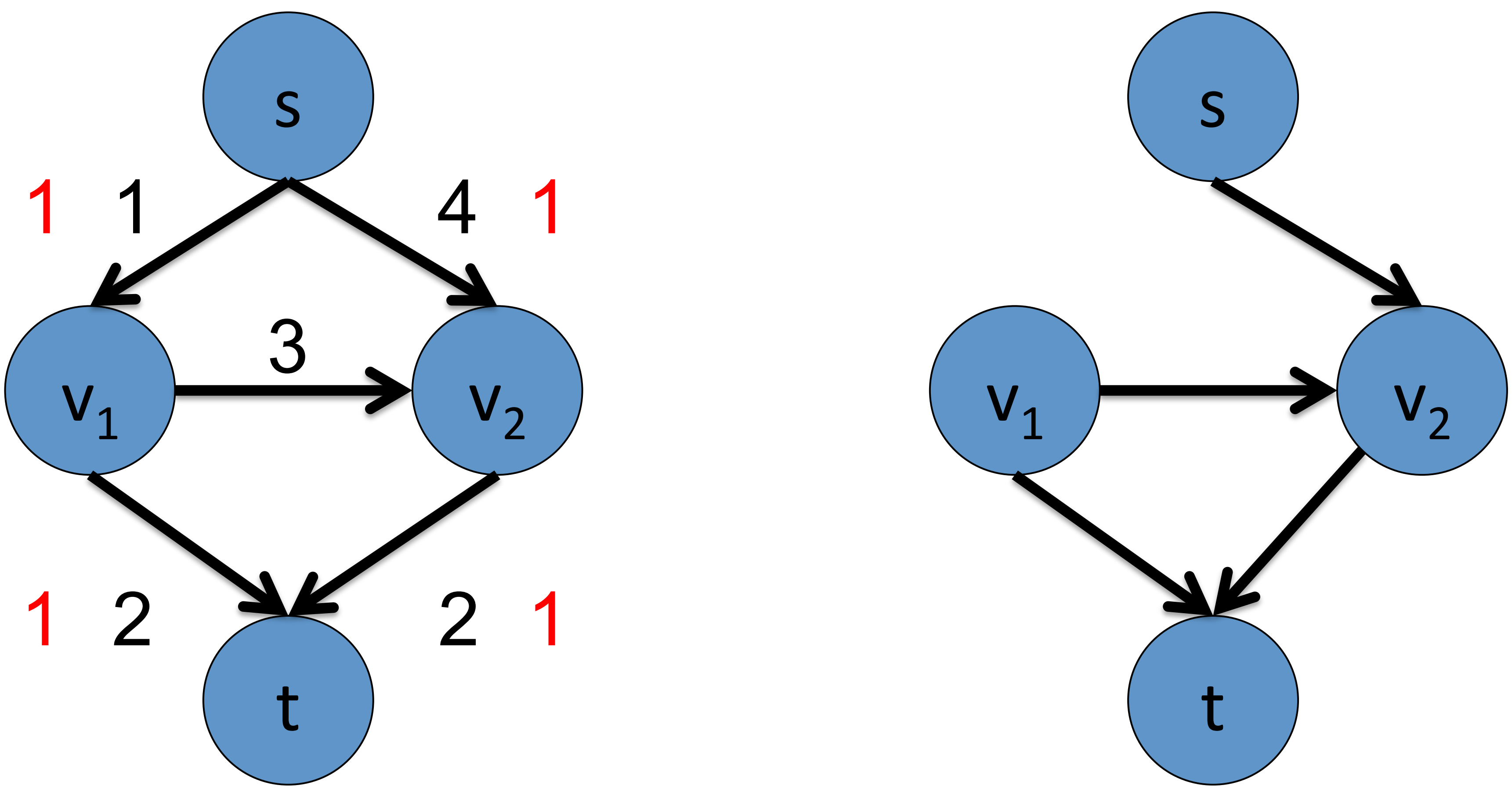
Add K to (s,v_2) and (v_1,t). Subtract K from (v_1,v_2).

Maximum Flow using Residual Graphs



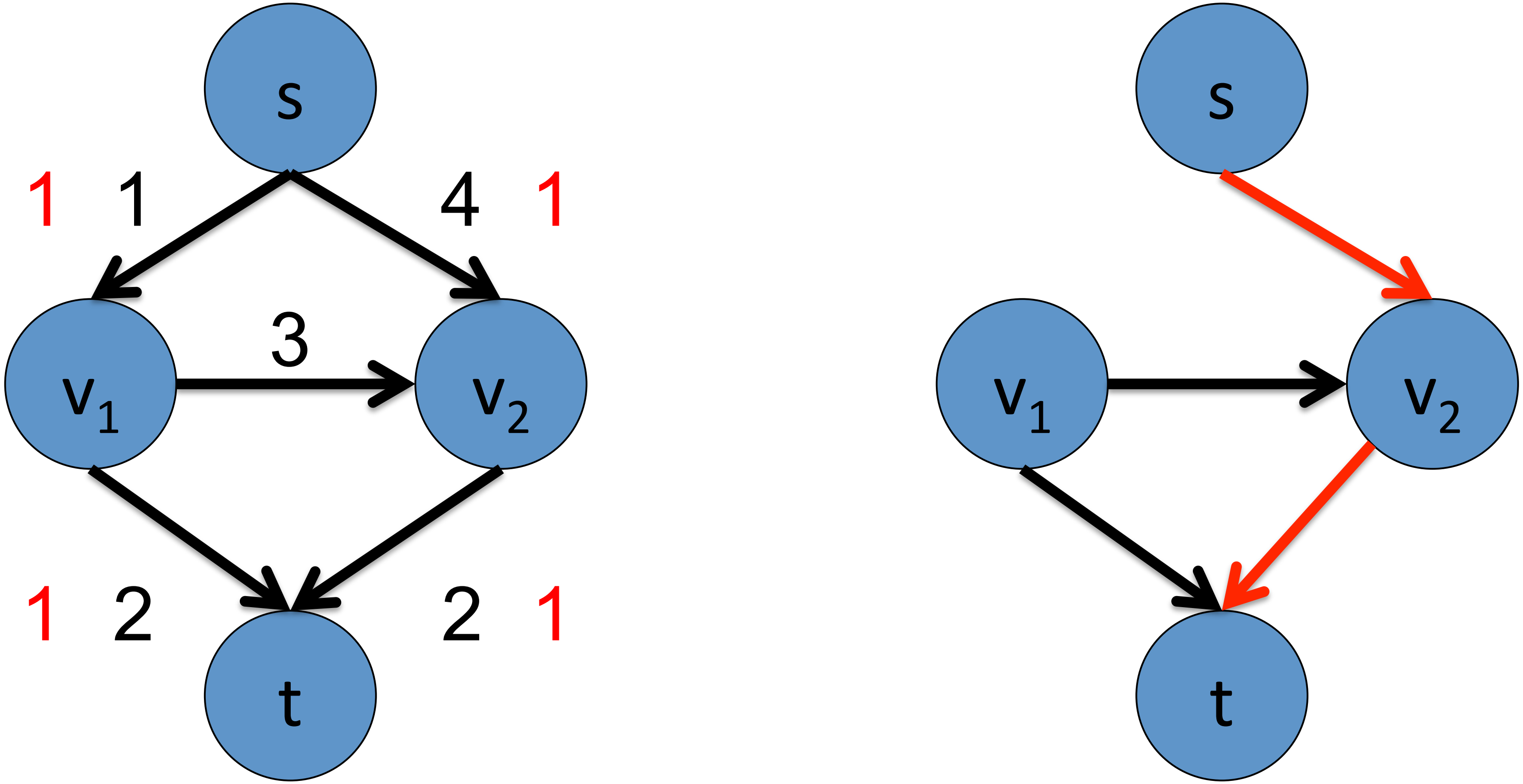
Update the residual graph.

Maximum Flow using Residual Graphs



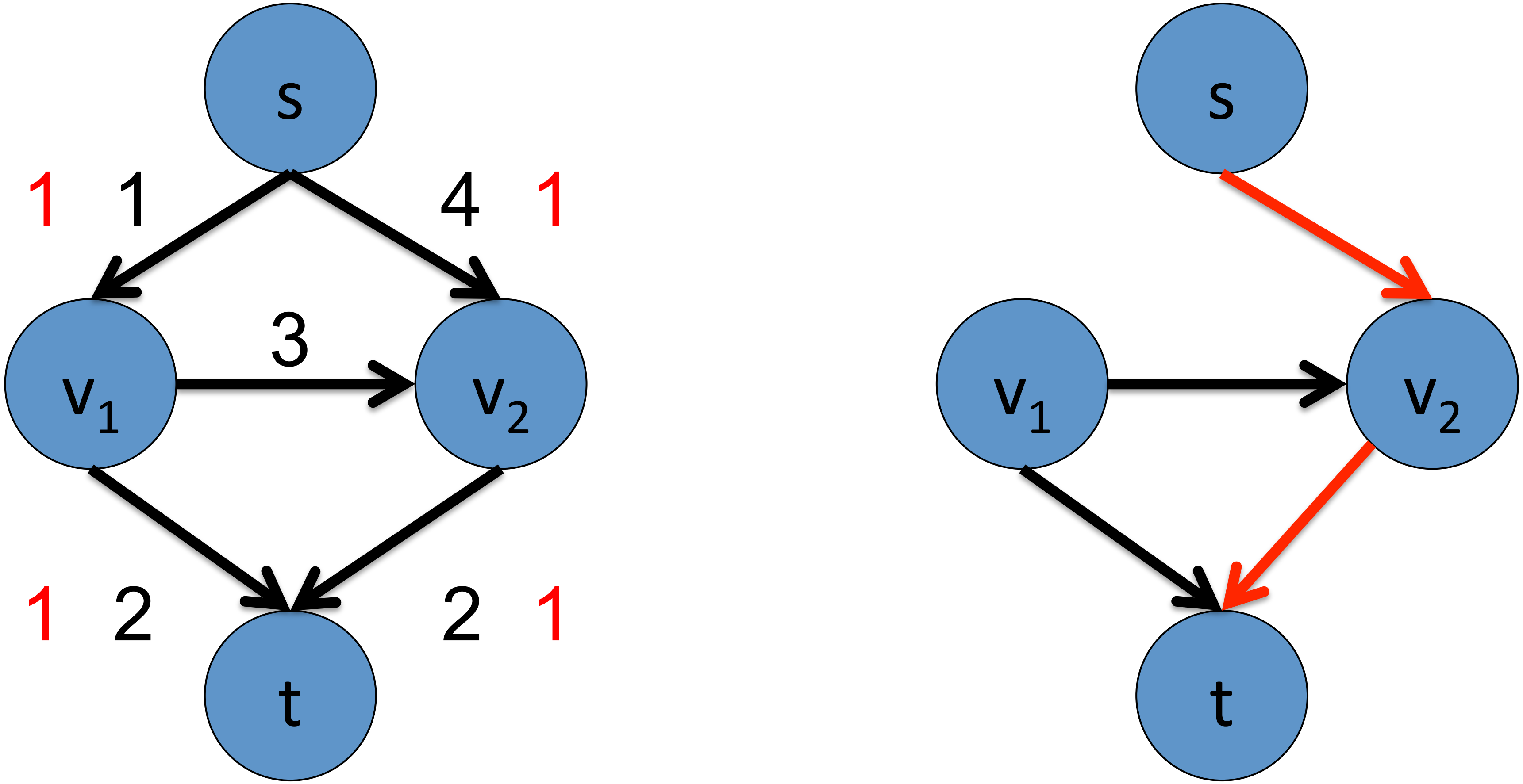
Find an s-t path in the residual graph.

Maximum Flow using Residual Graphs



Find an s-t path in the residual graph.

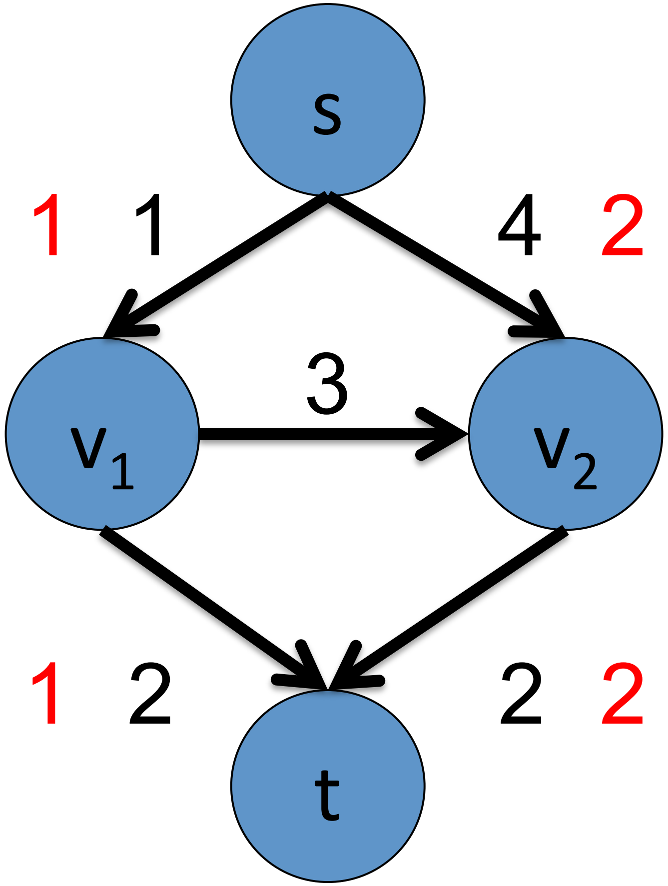
Maximum Flow using Residual Graphs

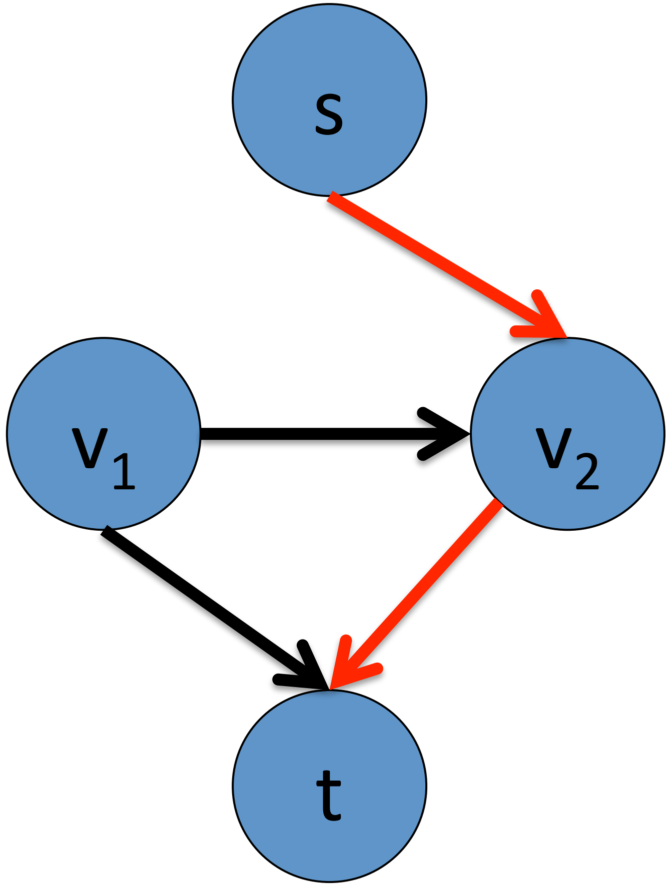


Choose maximum allowable value of K.

Add K to (s,v_2) and (v_2,t).

Maximum Flow using Residual Graphs

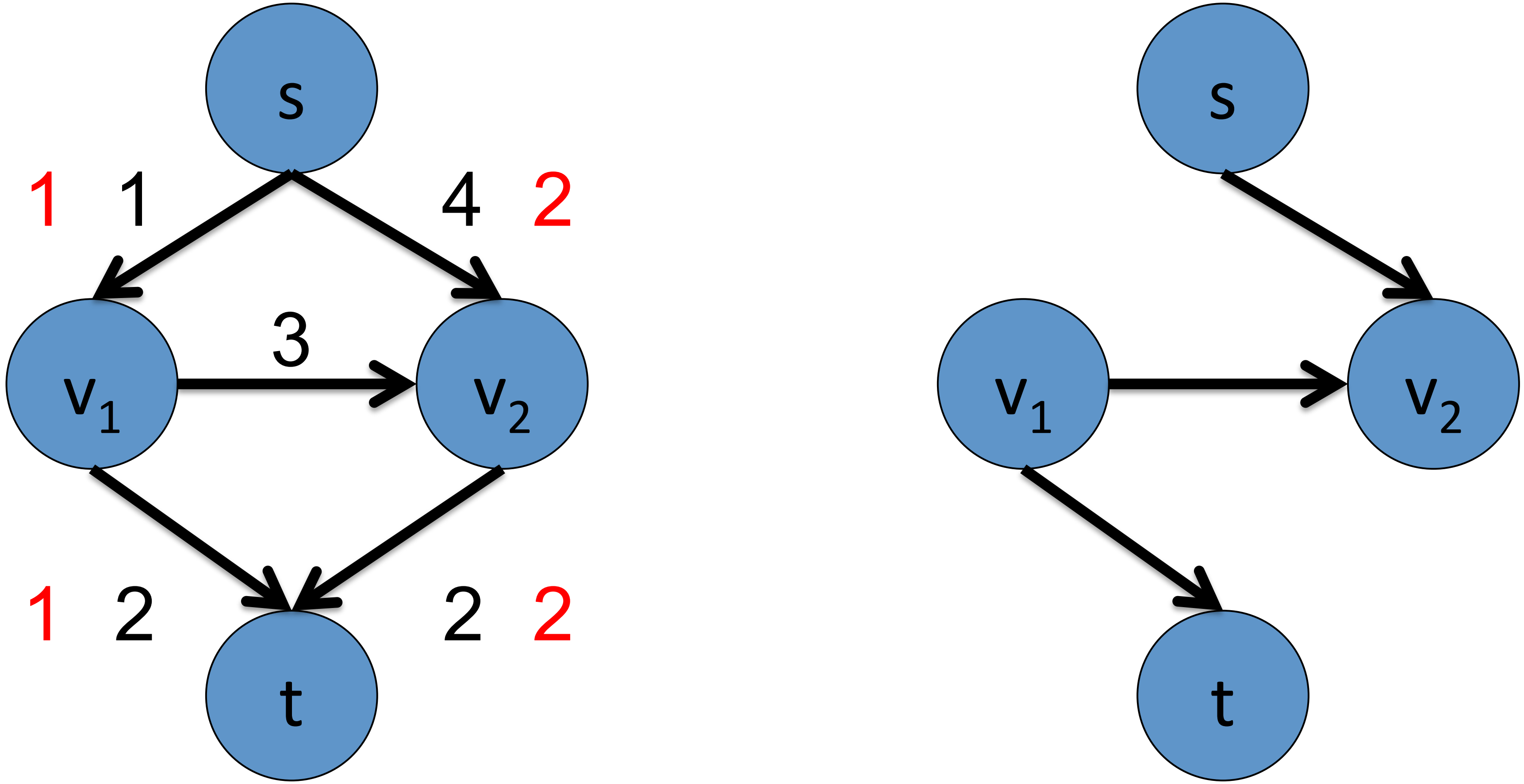




Choose maximum allowable value of K.

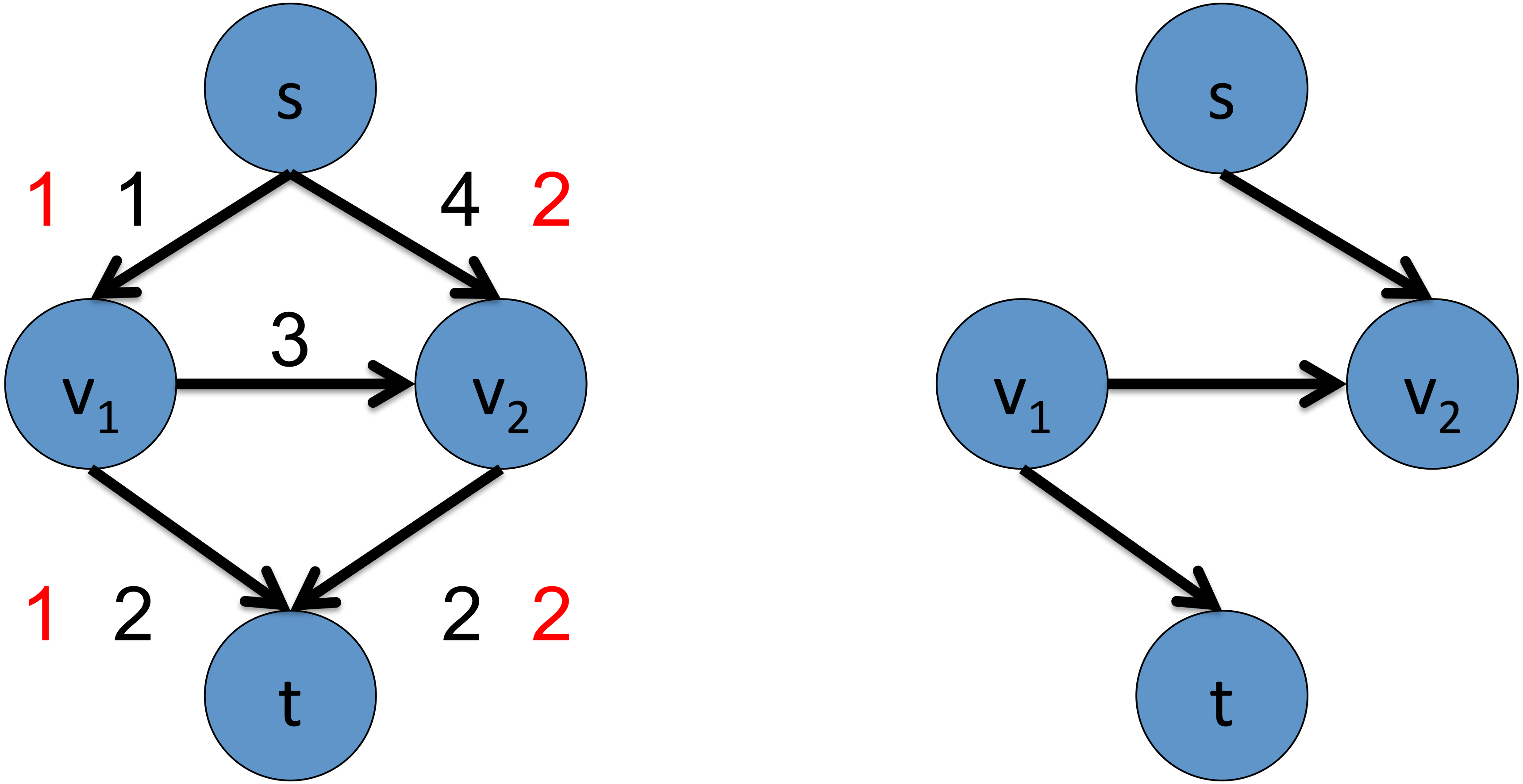
Add K to (s,v_2) and (v_2,t).

Maximum Flow using Residual Graphs



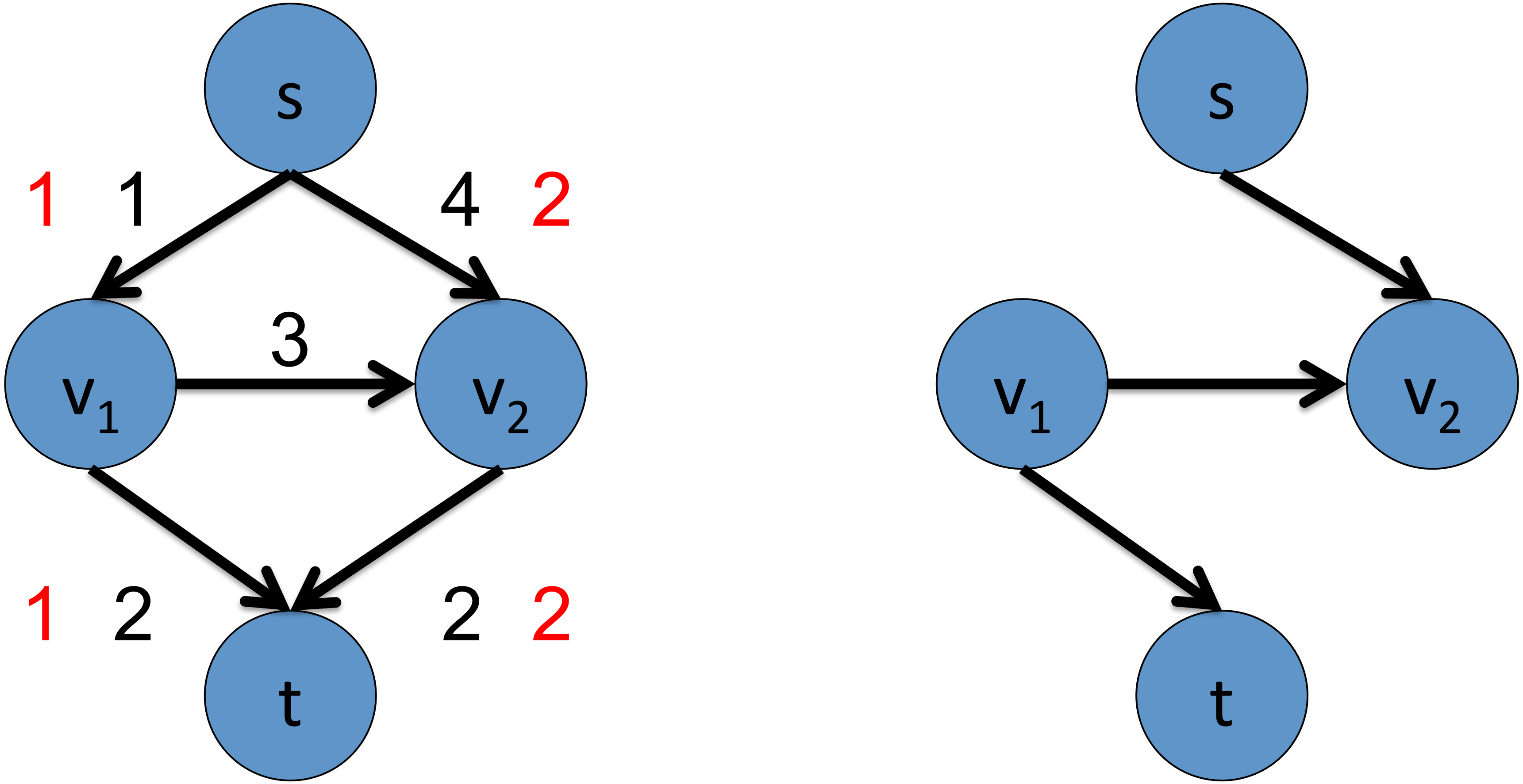
Update residual graph.

Maximum Flow using Residual Graphs



No more s-t paths. Stop.

Maximum Flow using Residual Graphs

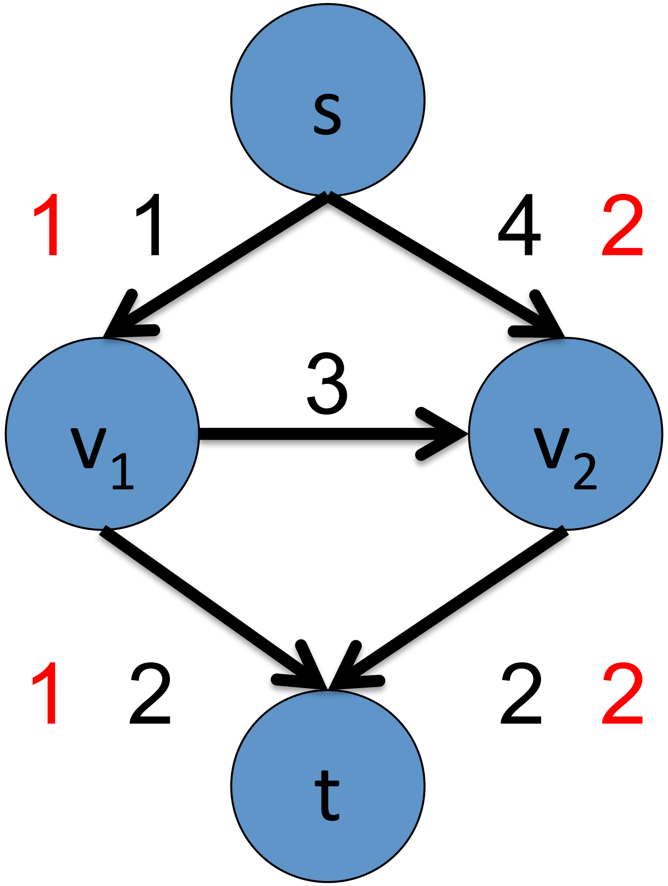


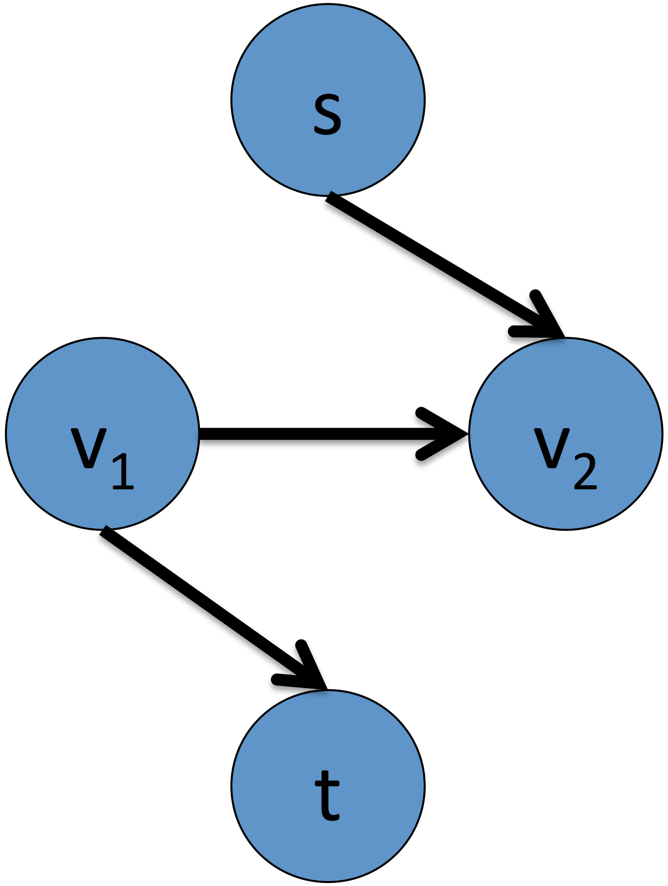
How can I be sure this will always work?

Outline

- Preliminaries
- Maximum Flow
  - Residual Graph
  - **Max-Flow Min-Cut Theorem**
- Algorithms

Max-Flow Min-Cut

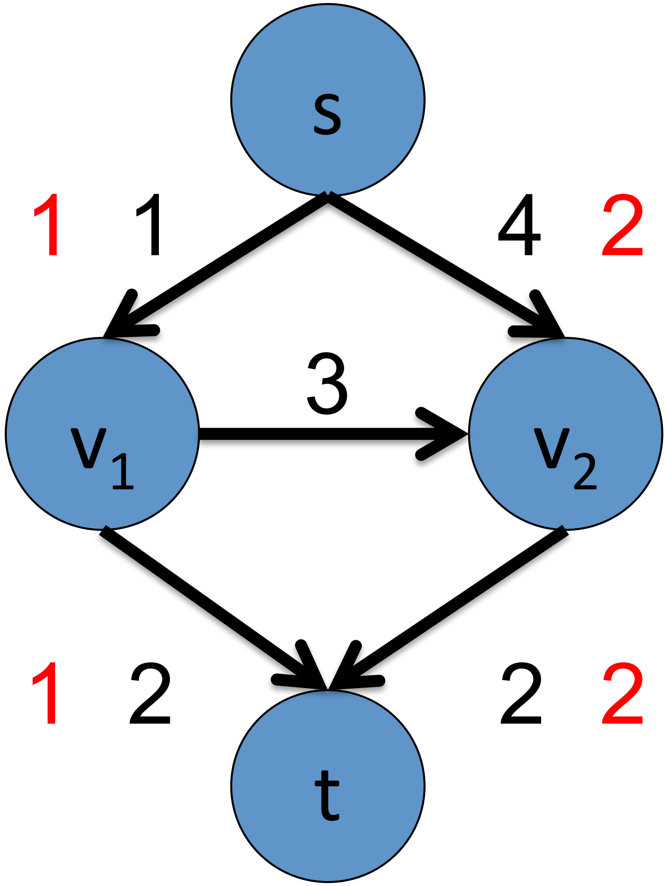


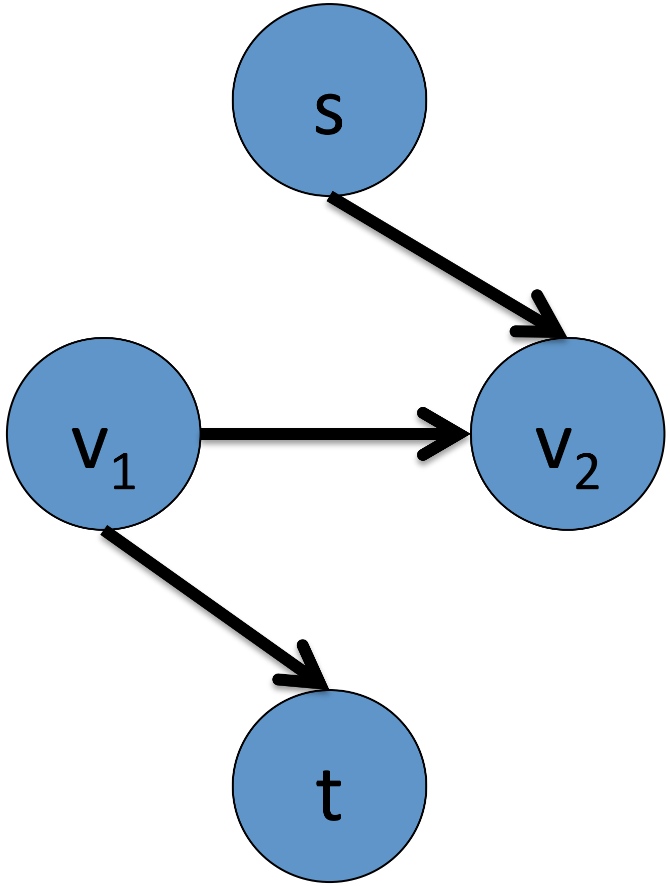


t is not in U.

Let the subset of vertices U be reachable from s.

Max-Flow Min-Cut

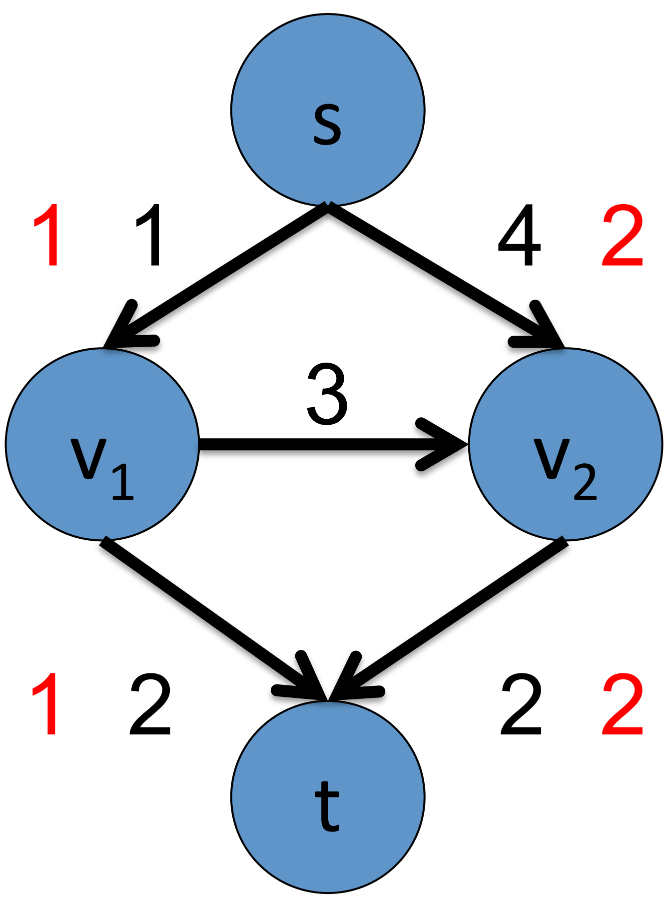


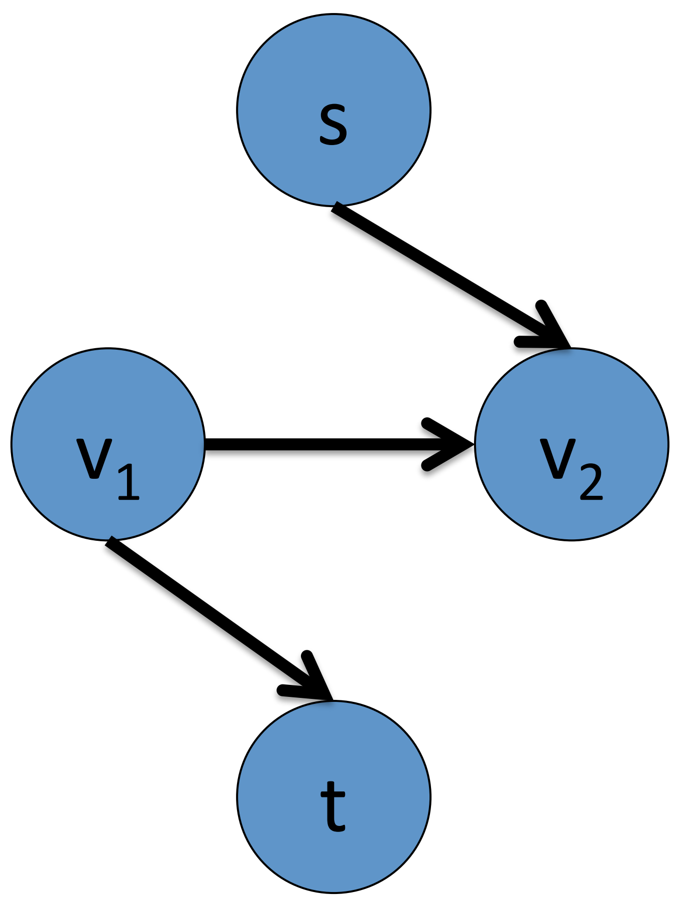


Or else a will be in the residual graph.

For all a ∈ out-arcs(U), flow(a) = c(a).

Max-Flow Min-Cut

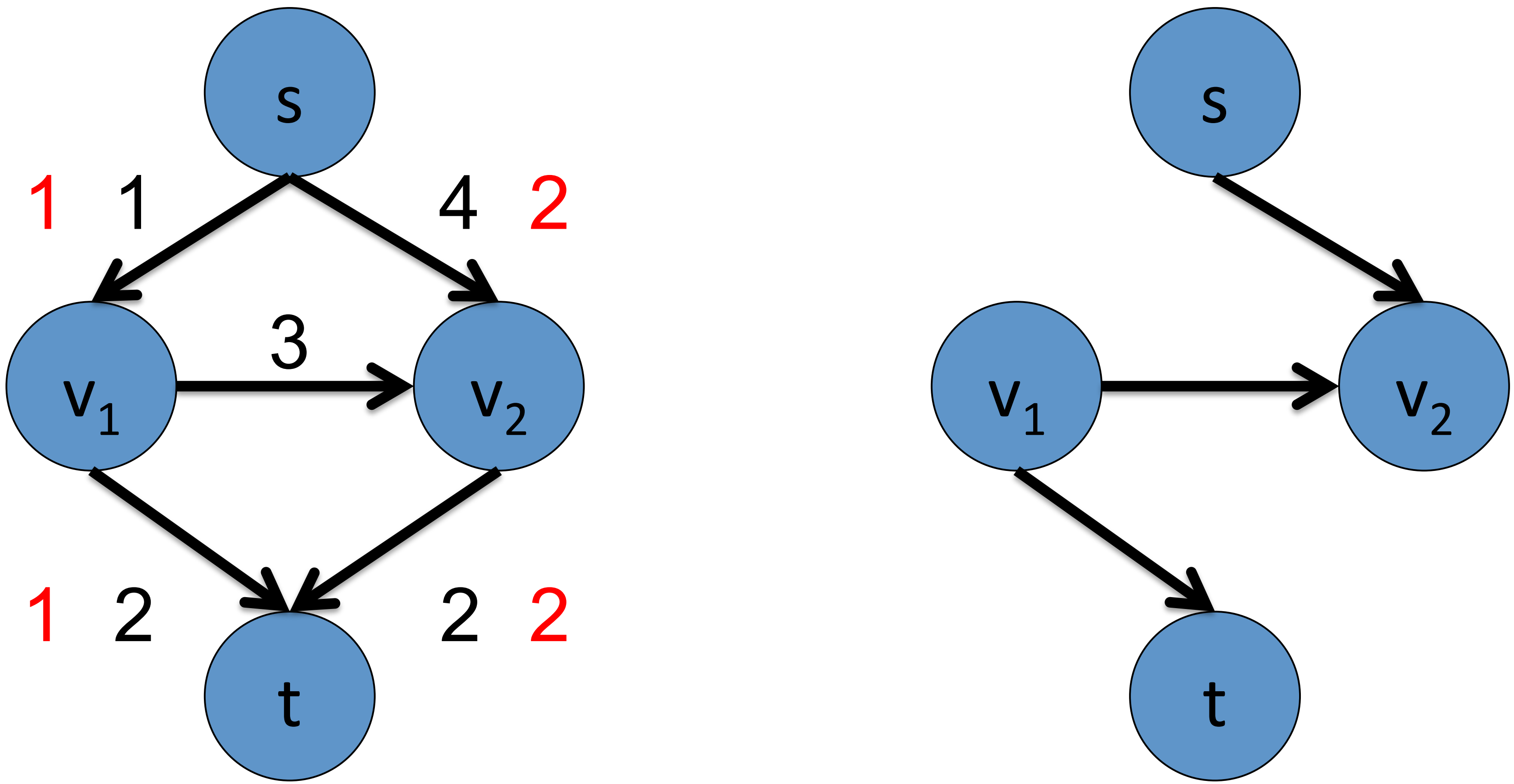




Or else inverse of a will be in the residual graph.

For all a ∈ in-arcs(U), flow(a) = 0.

Max-Flow Min-Cut



For all a ∈ out-arcs(U), flow(a) = c(a).

For all a ∈ in-arcs(U), flow(a) = 0.

Flows vs. Cuts

Value of flow $= -E_{flow}(s)$

$= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$

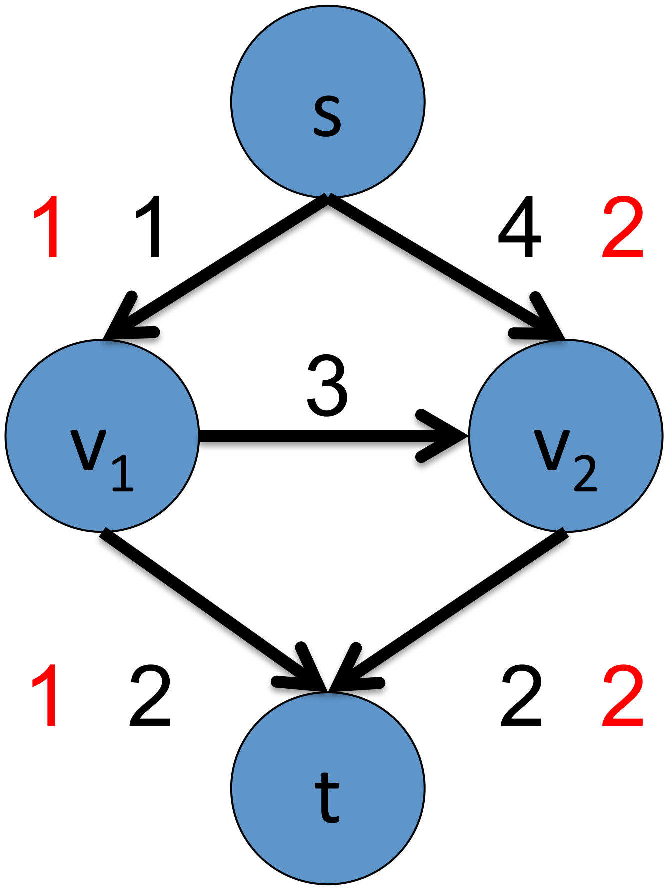
$= -E_{flow}(U)$

= flow(out-arcs(U))
 - flow(in-arcs(U))

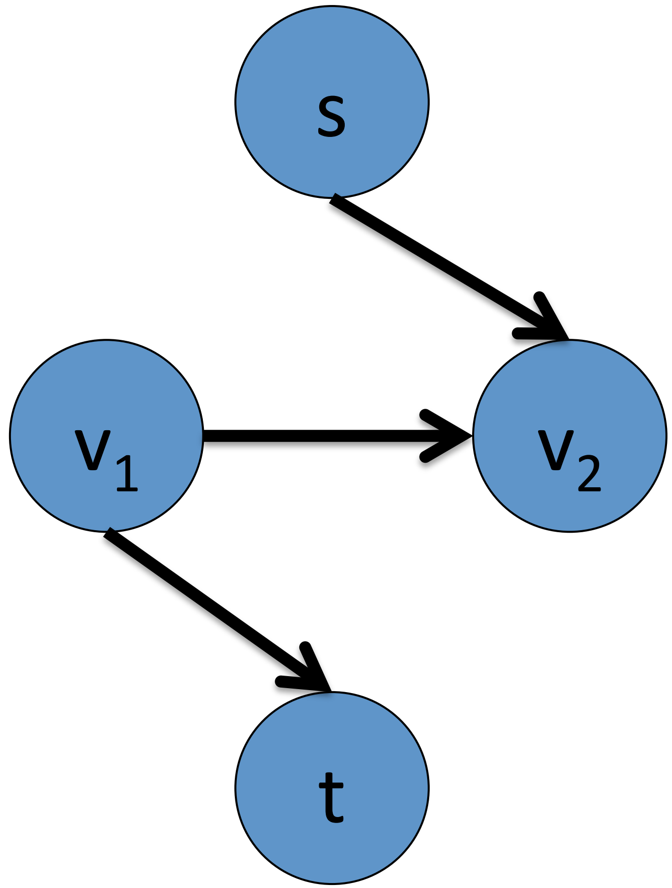
= Capacity of C

flow(a) = c(a), a $\in$ out-arcs(U)   flow(a) = 0, a $\in$ in-arcs(U)

Max-Flow Min-Cut



Minimum Cut



Maximum Flow

Capacity(C) = Value(flow)

Outline

- Preliminaries
- Maximum Flow
- **Algorithms**
  - **Ford-Fulkerson Algorithm**
  - Dinits Algorithm

Ford-Fulkerson Algorithm

Start with flow = 0 for all arcs.

Find an s-t path in the residual graph.

Pass maximum allowable flow.

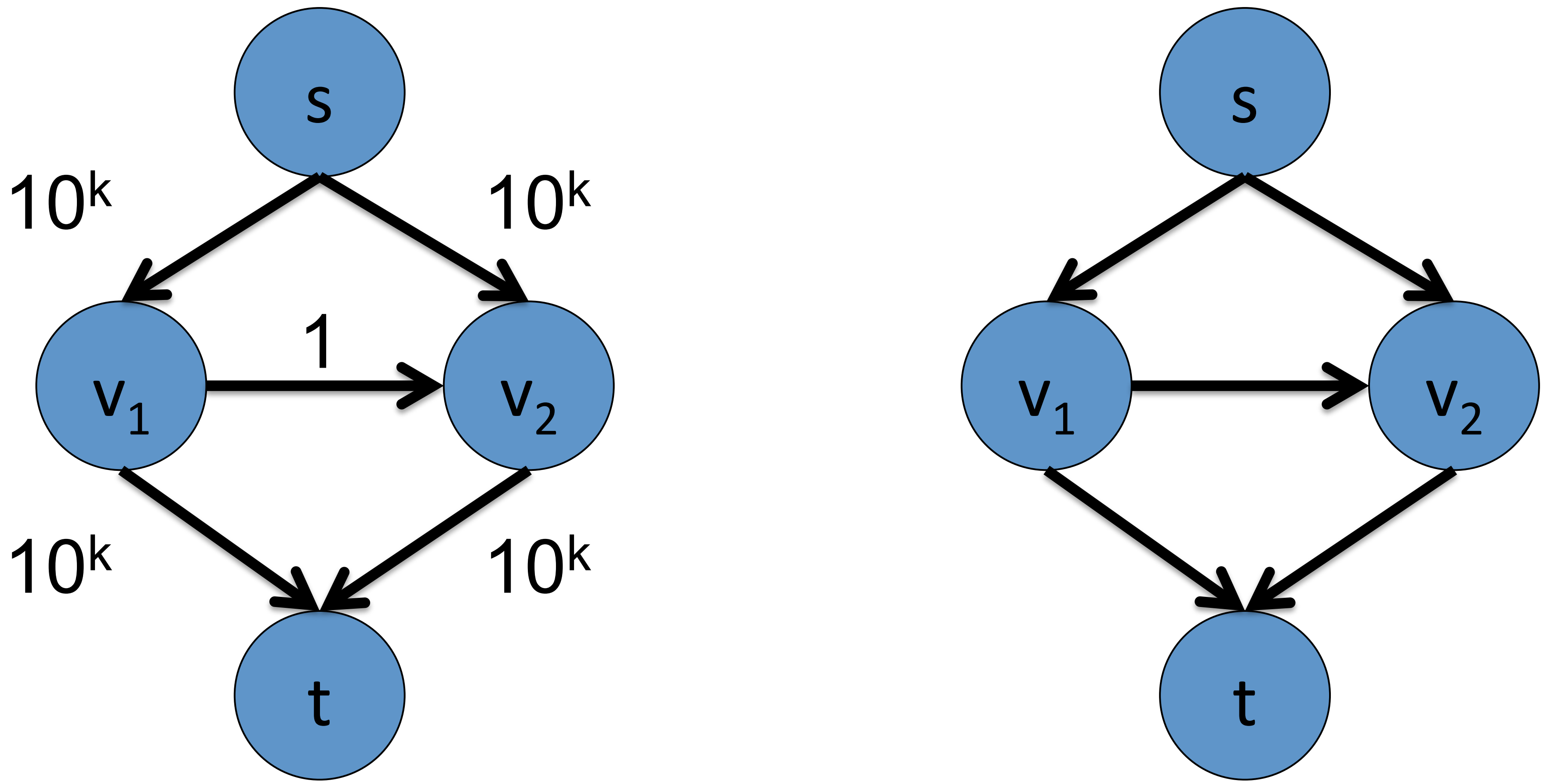
Subtract from inverse arcs.

Add to forward arcs.

REPEAT

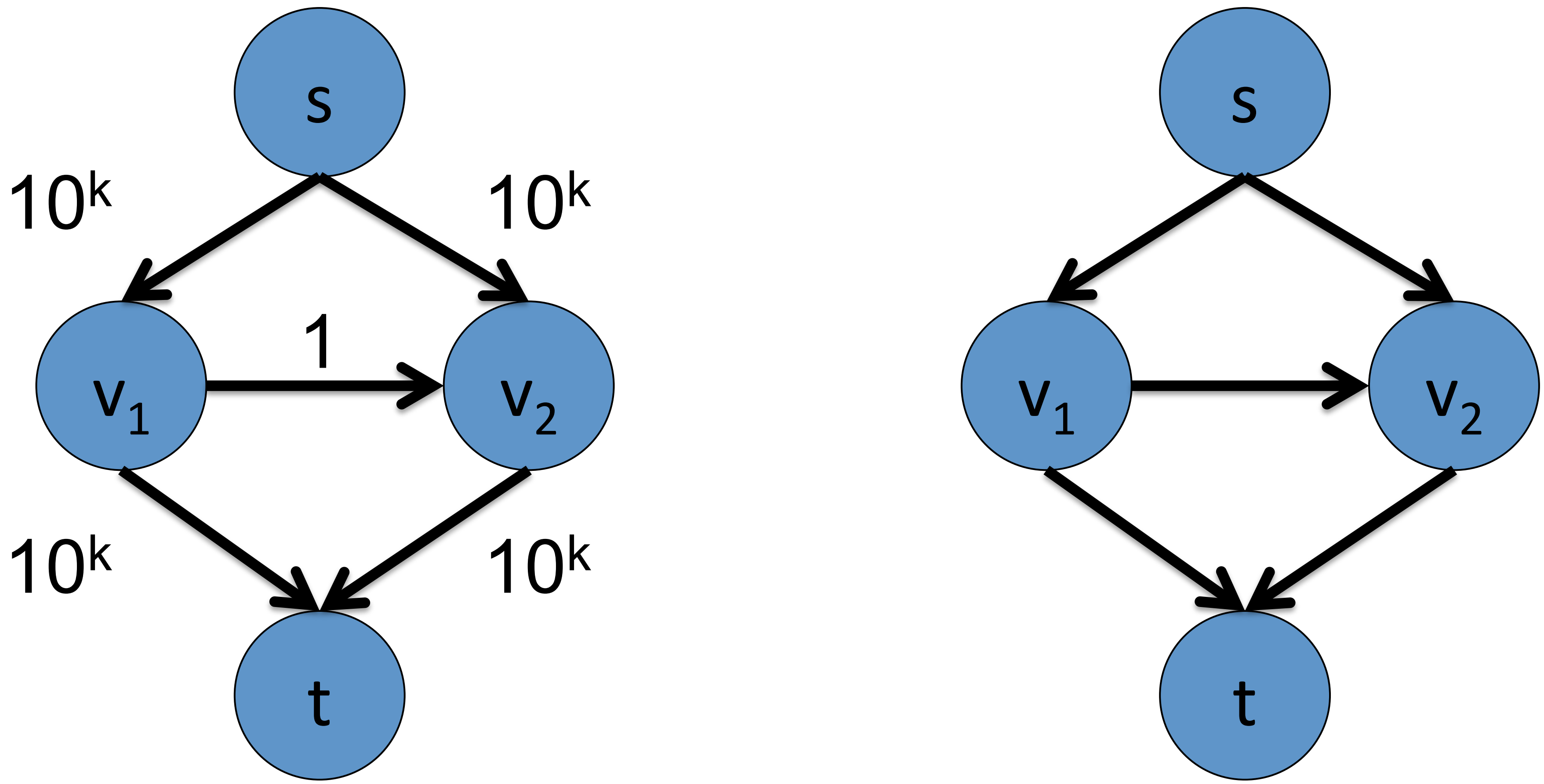
Until s and t are disjoint in the residual graph.

Ford-Fulkerson Algorithm



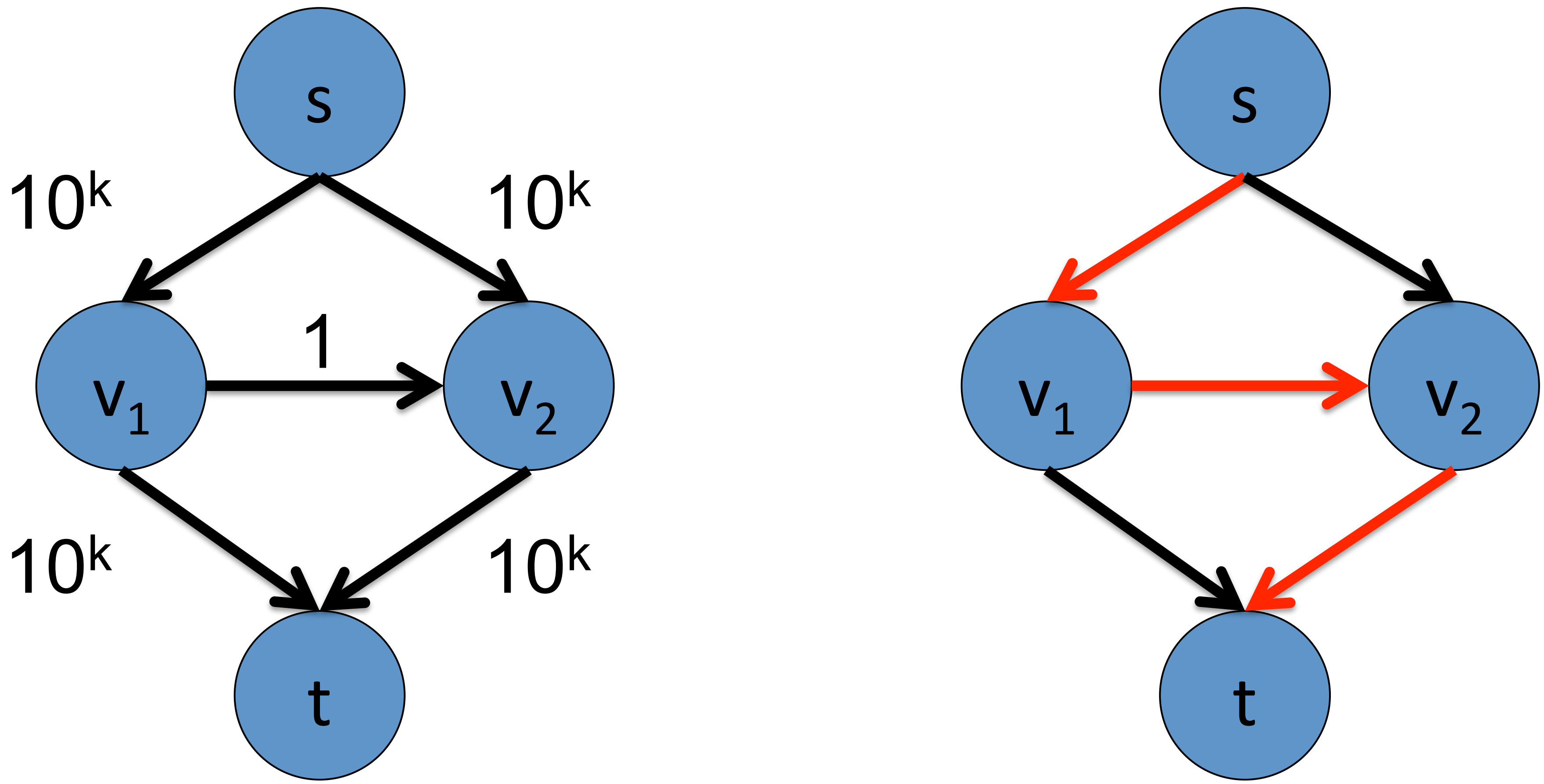
Start with zero flow

Ford-Fulkerson Algorithm



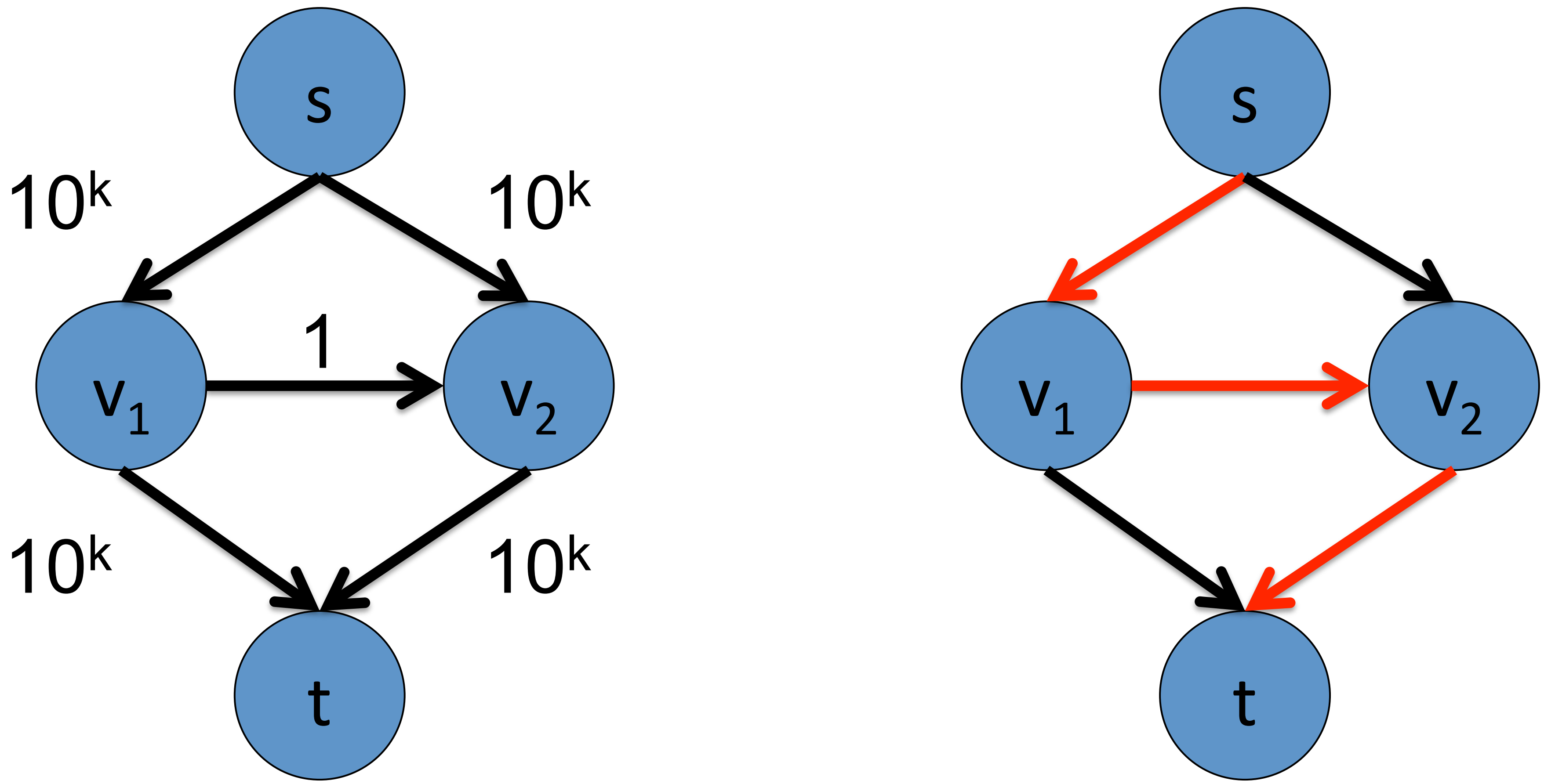
Find an s-t path in the residual graph.

Ford-Fulkerson Algorithm



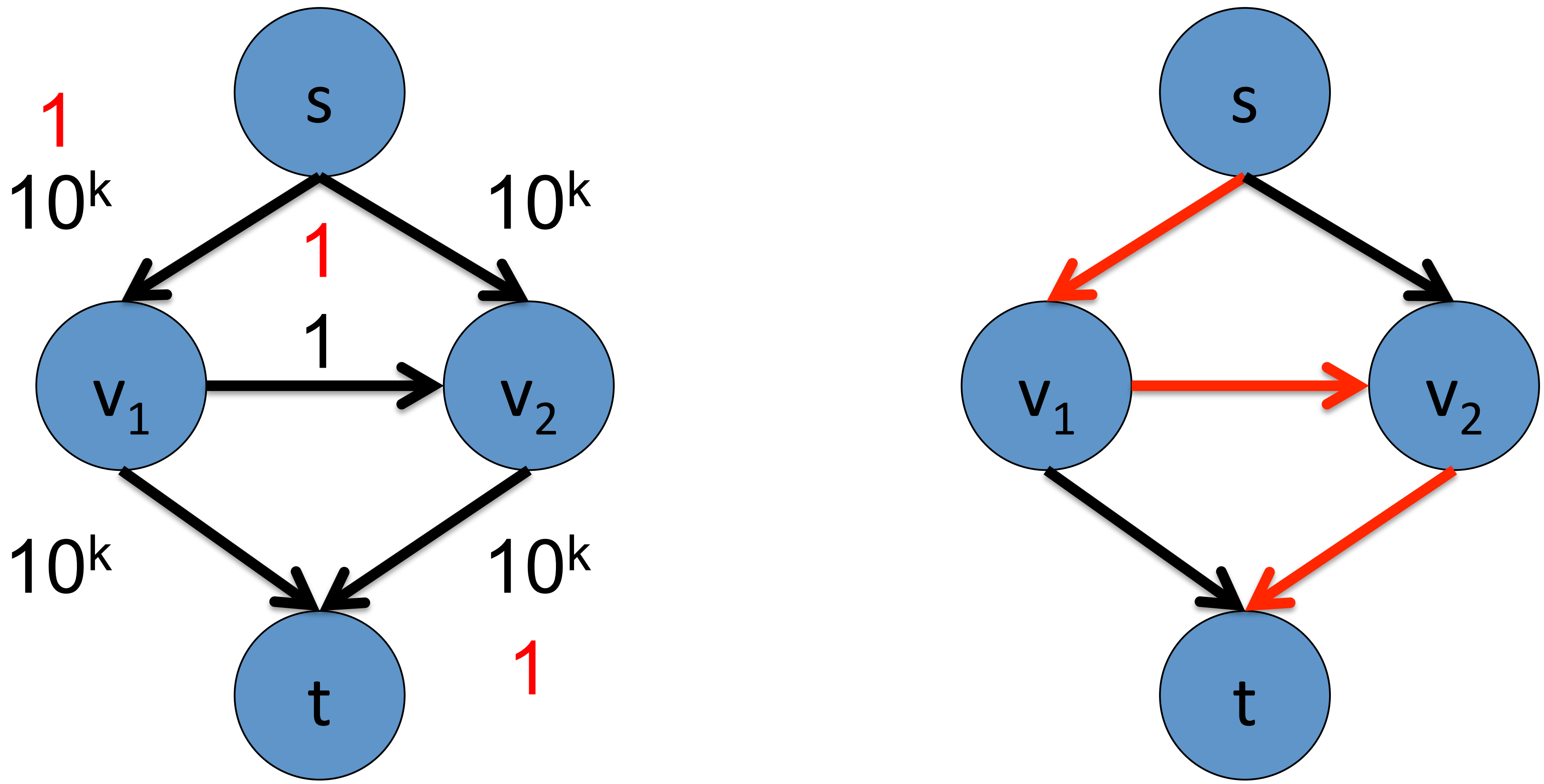
Find an s-t path in the residual graph.

Ford-Fulkerson Algorithm



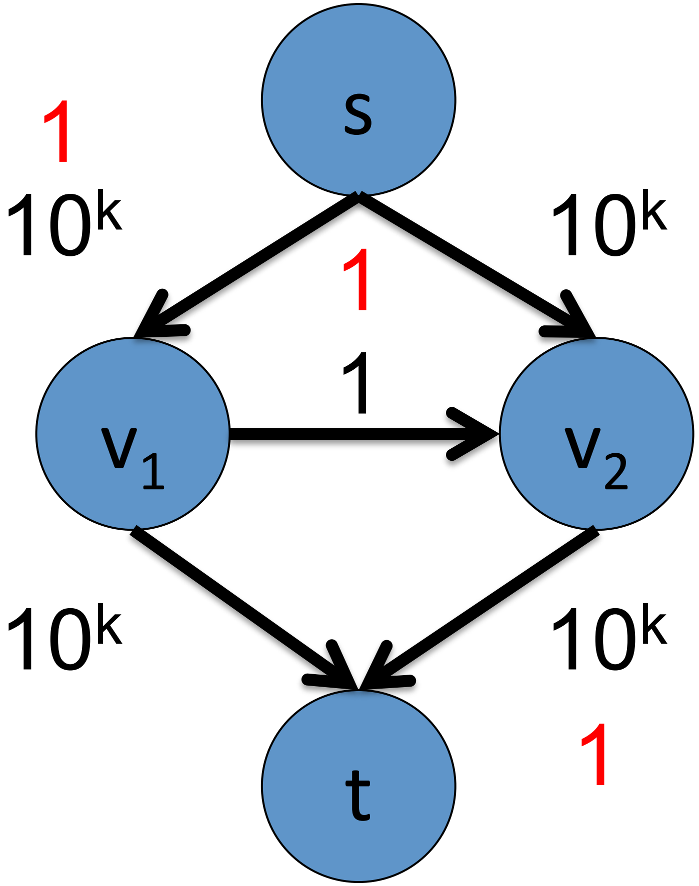
Pass the maximum allowable flow.

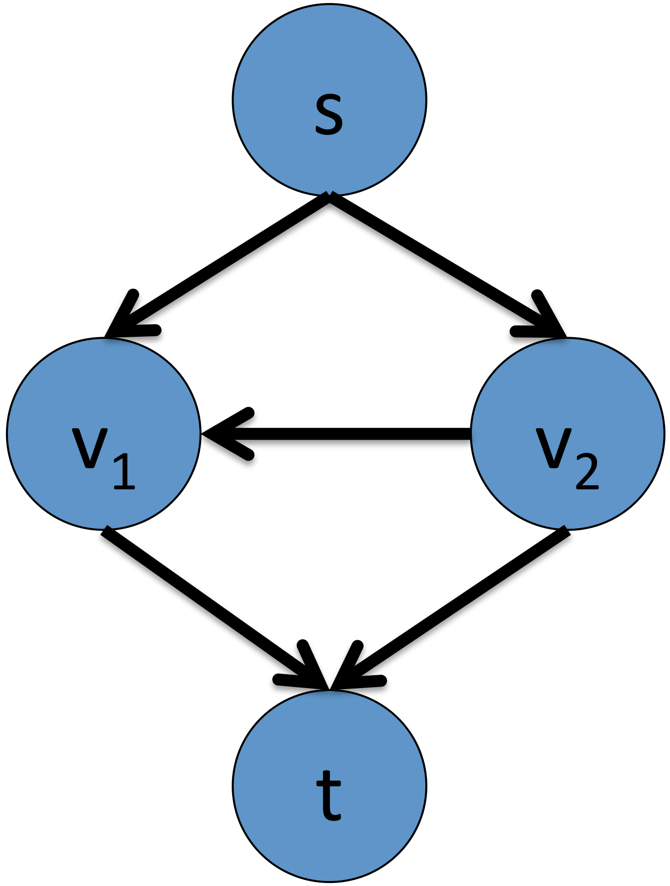
Ford-Fulkerson Algorithm



Pass the maximum allowable flow.

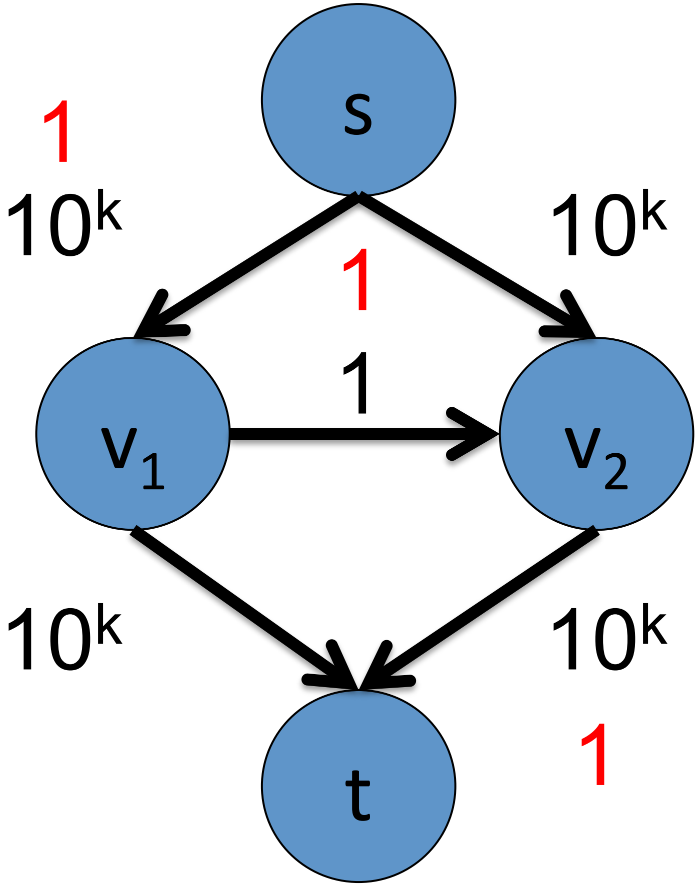
Ford-Fulkerson Algorithm

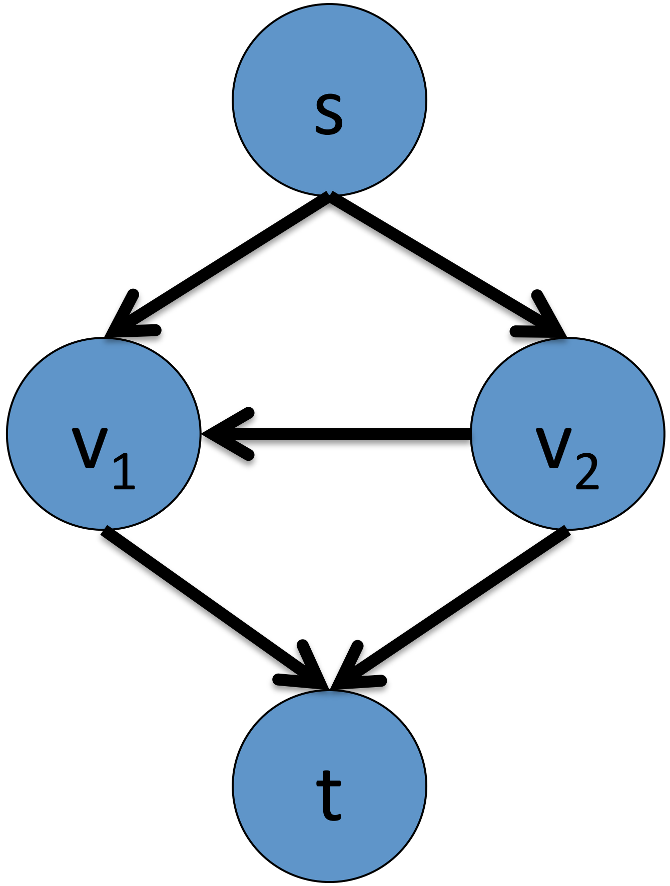




Update the residual graph.

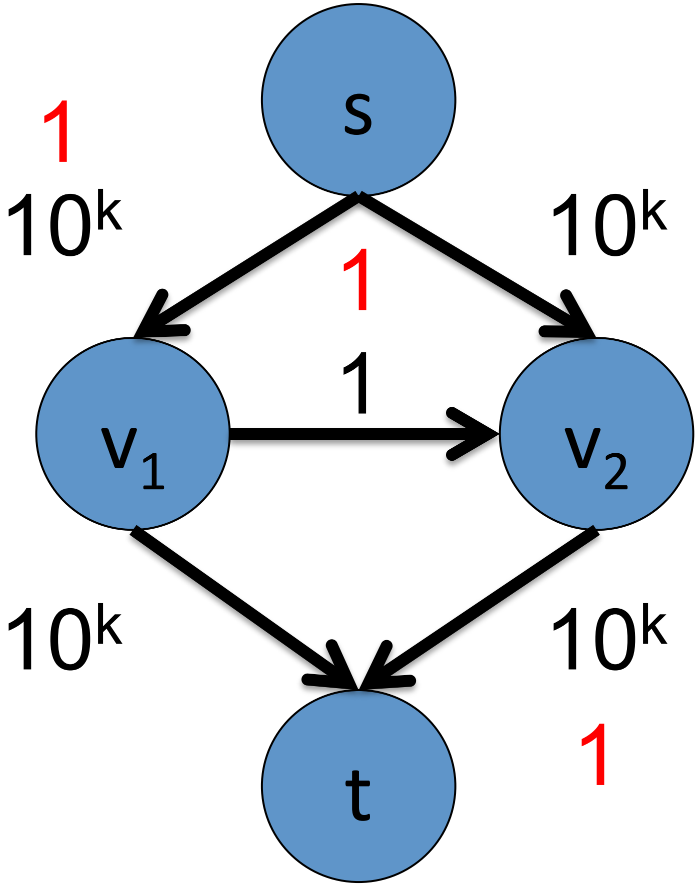
Ford-Fulkerson Algorithm

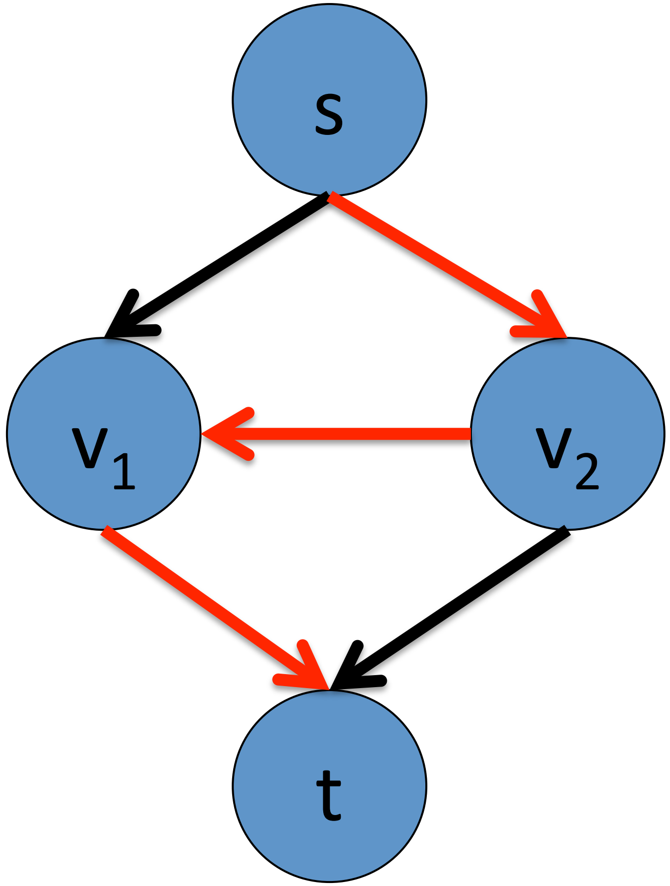




Find an s-t path in the residual graph.

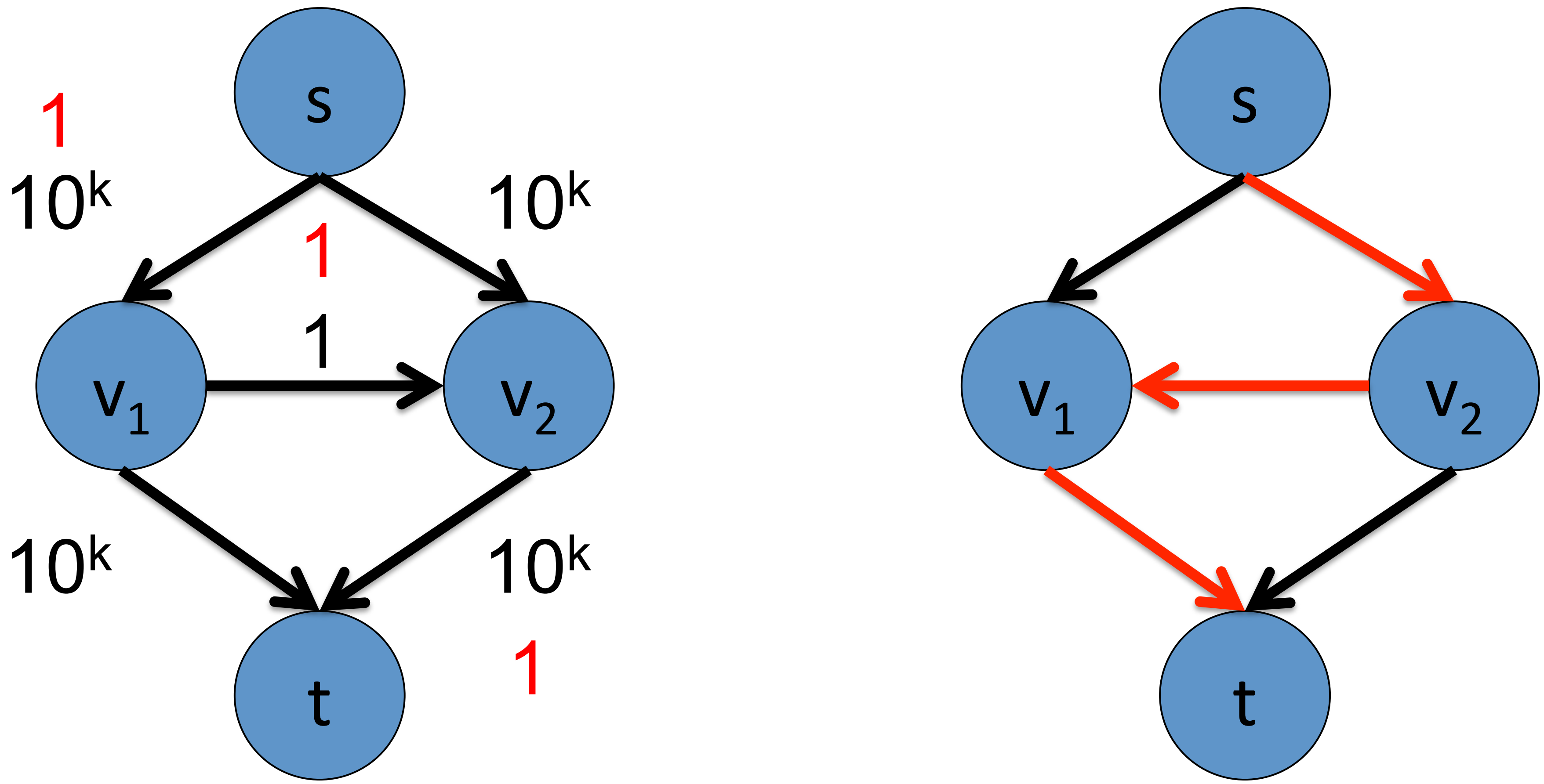
Ford-Fulkerson Algorithm





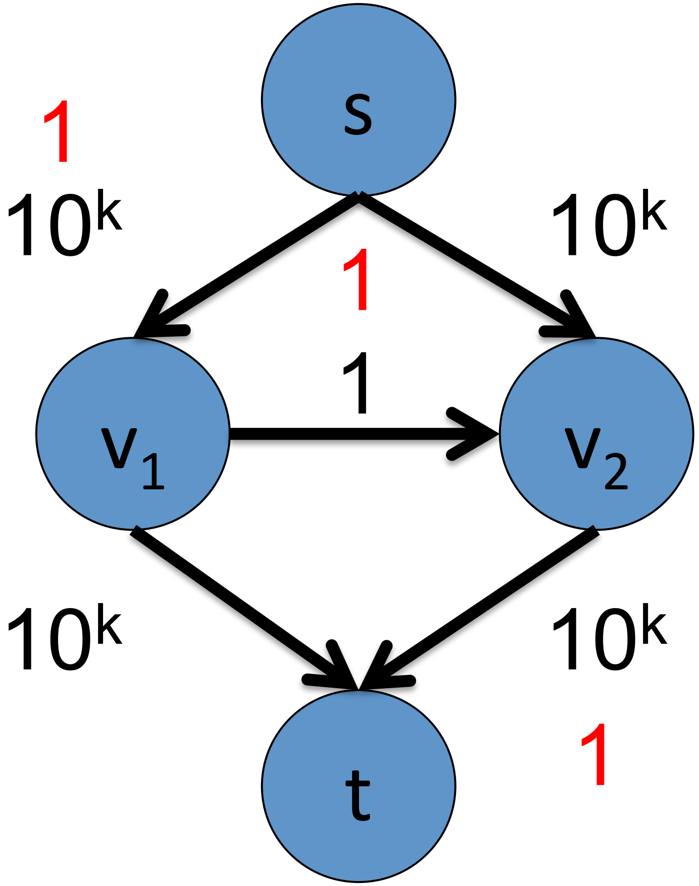
Find an s-t path in the residual graph.

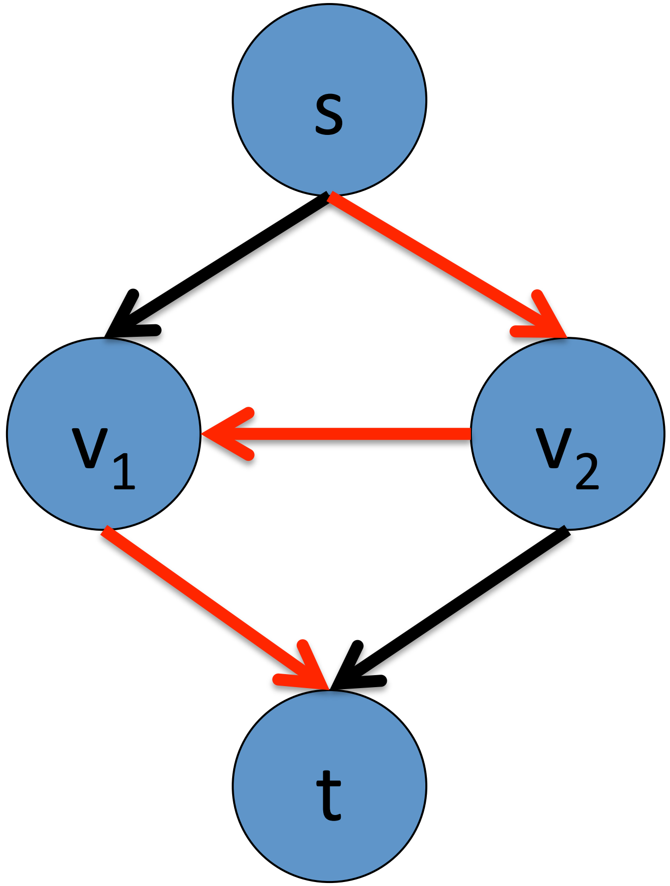
Ford-Fulkerson Algorithm



Complexity is exponential in k.

Ford-Fulkerson Algorithm

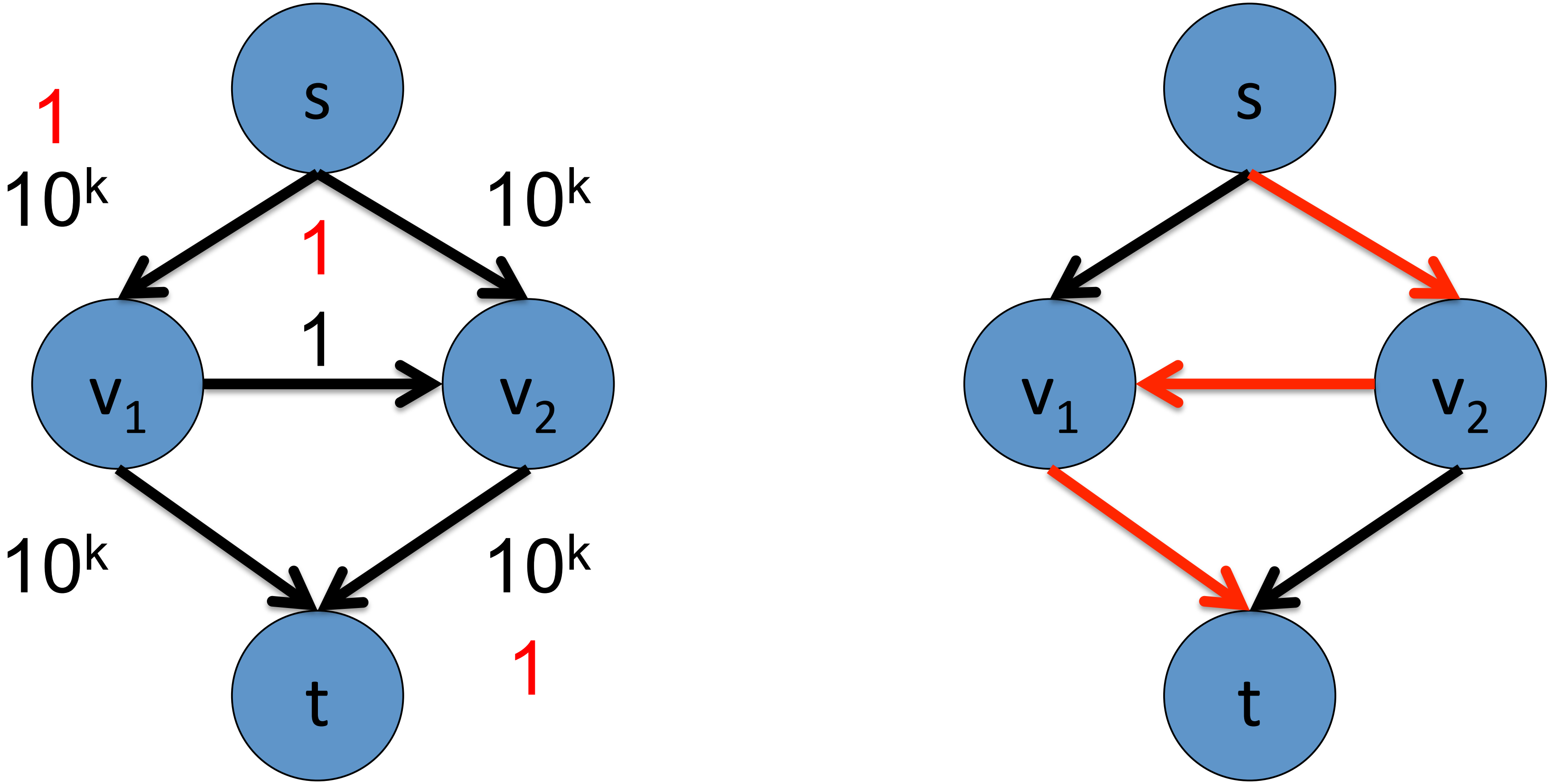




For examples, see Uri Zwick, 1993

Irrational arc lengths can lead to infinite iterations.

Ford-Fulkerson Algorithm



Choose wisely.

There are good paths and bad paths.

Outline

- Preliminaries
- Maximum Flow
- Algorithms
  - Ford-Fulkerson Algorithm
  - **Dinits Algorithm (Simple Version)**

Dinits Algorithm

Start with flow = 0 for all arcs.

Find the **minimum s-t path** in the residual graph.

Pass maximum allowable flow.

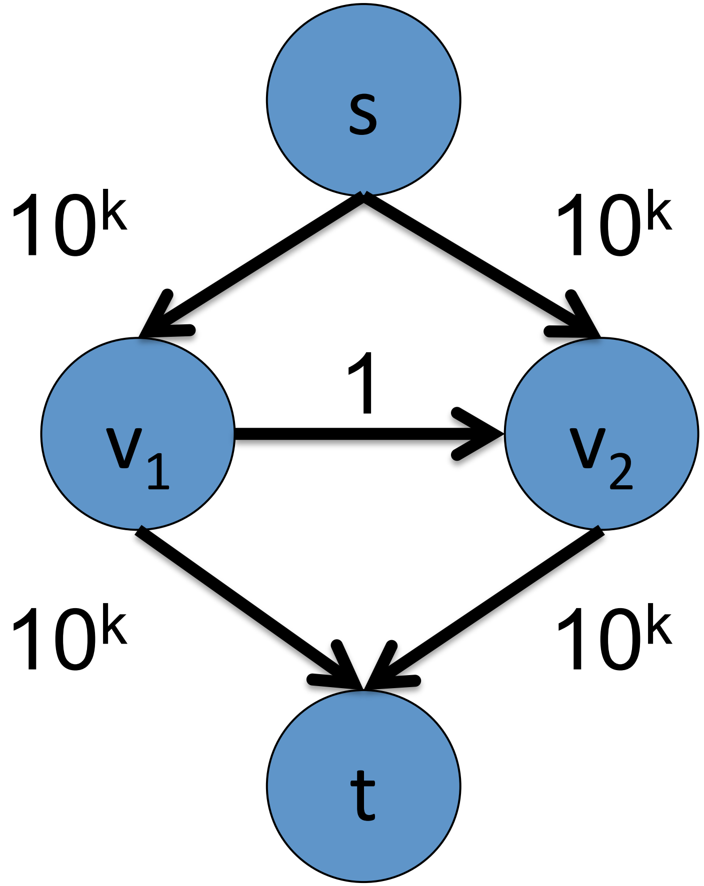
Subtract from inverse arcs.

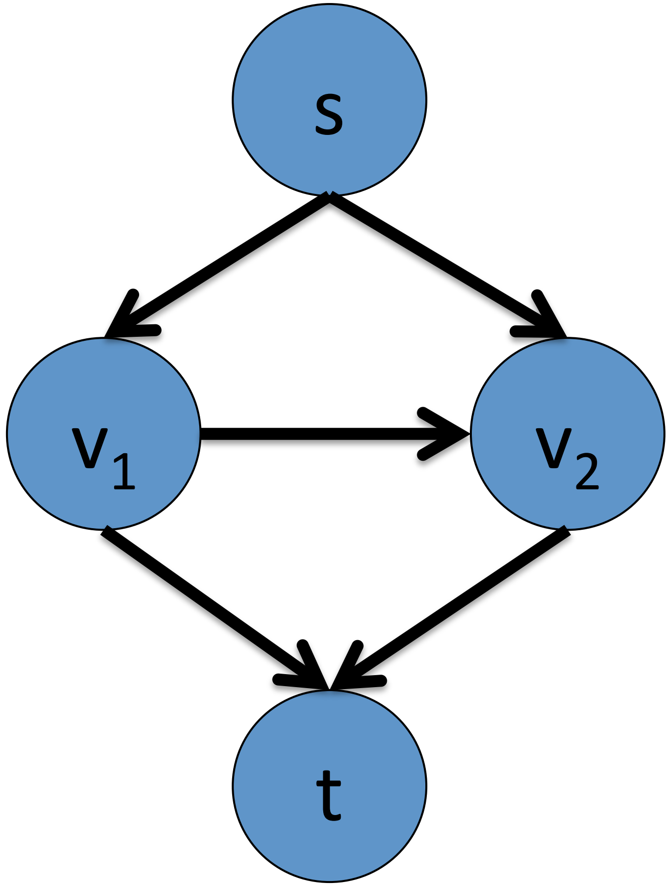
Add to forward arcs.

} REPEAT

Until s and t are disjoint in the residual graph.

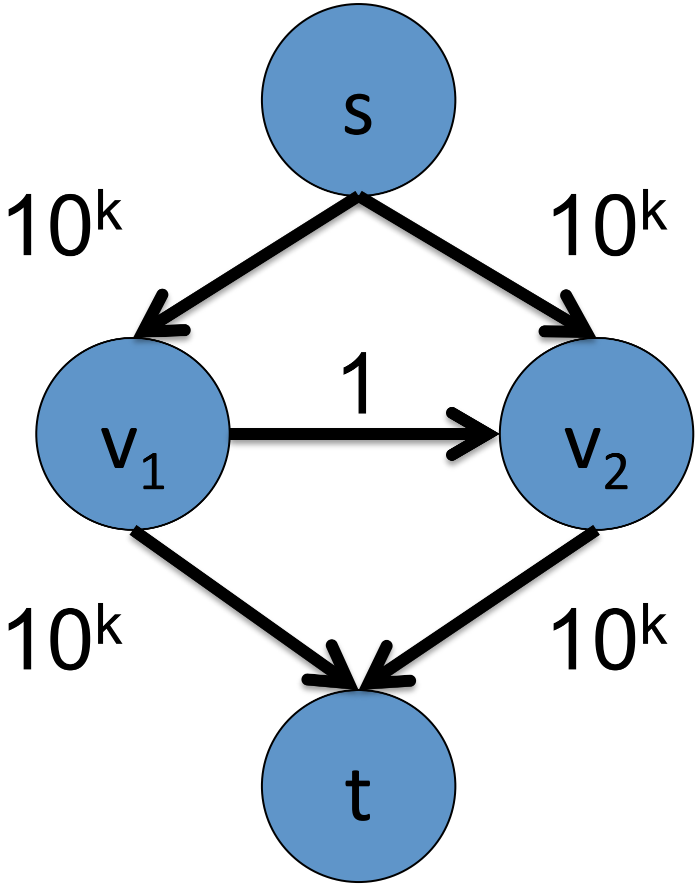
Dinits Algorithm

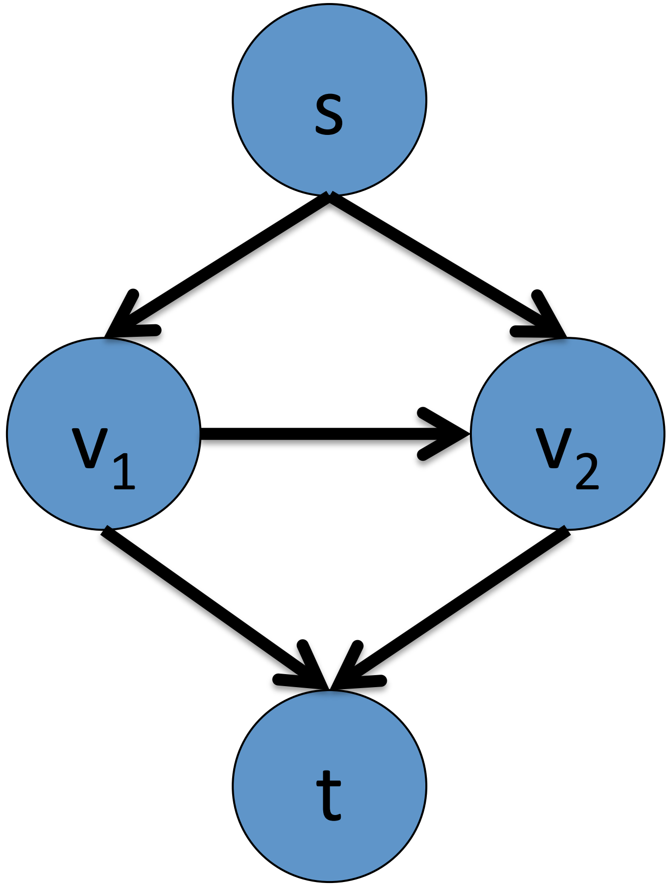




Start with zero flow

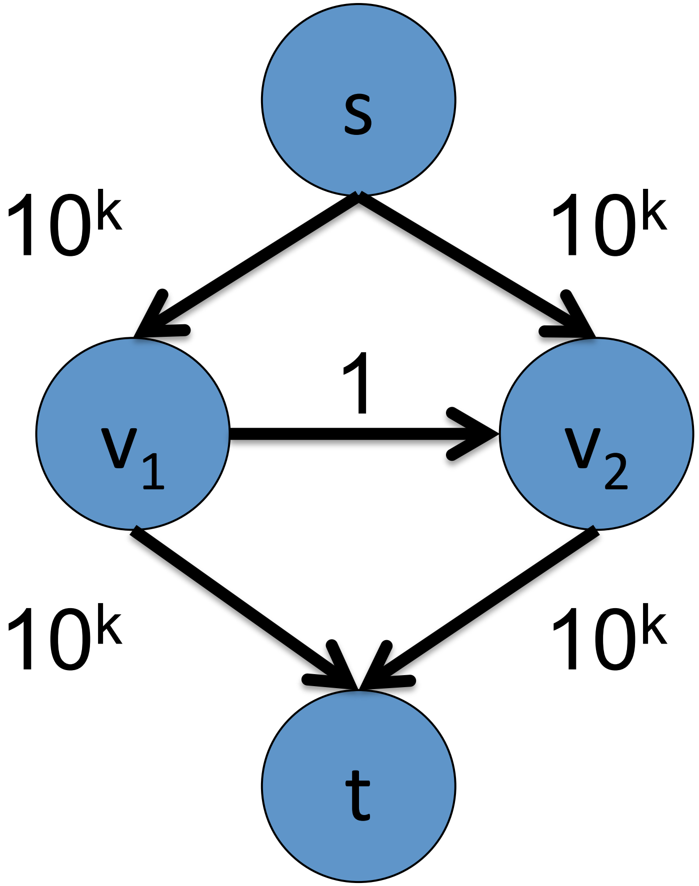
Dinits Algorithm

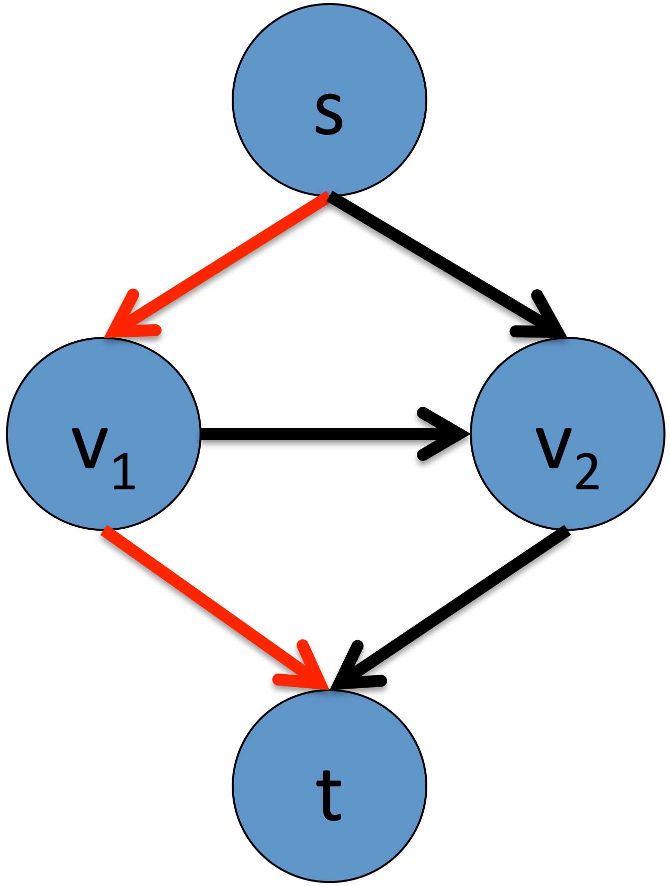




Find the minimum s-t path in the residual graph.

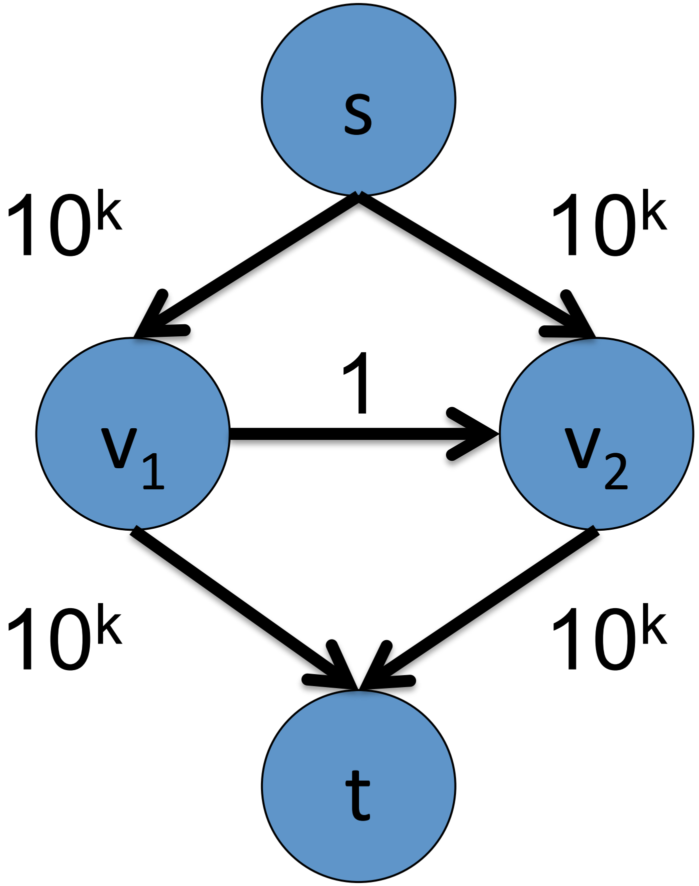
Dinits Algorithm

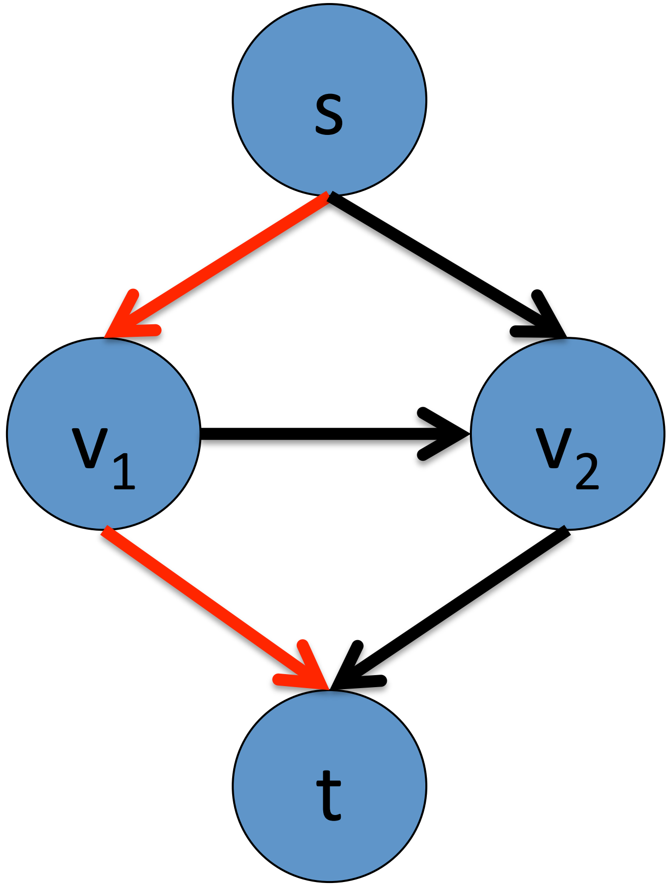




Find the minimum s-t path in the residual graph.

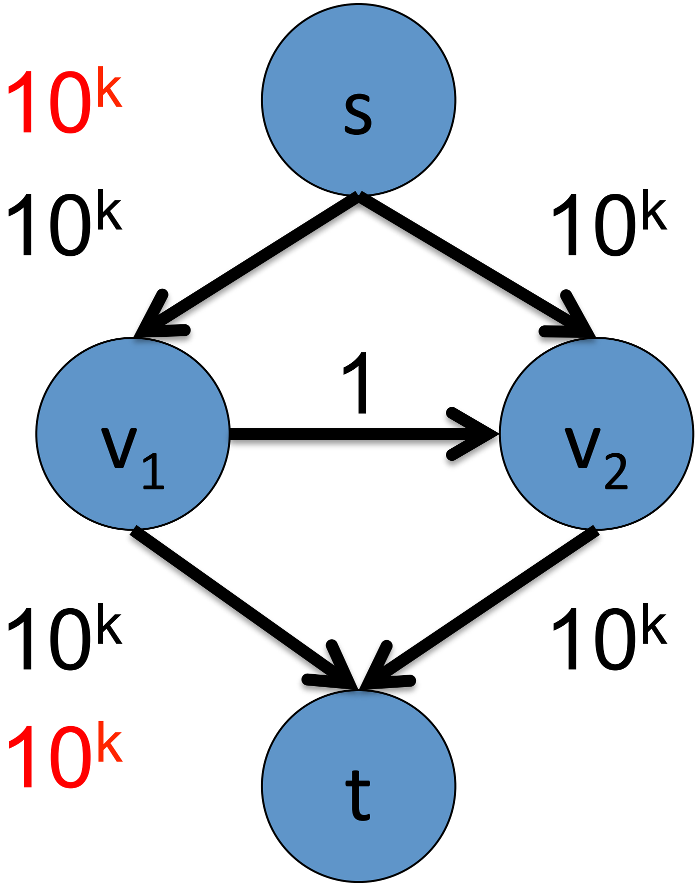
Dinits Algorithm

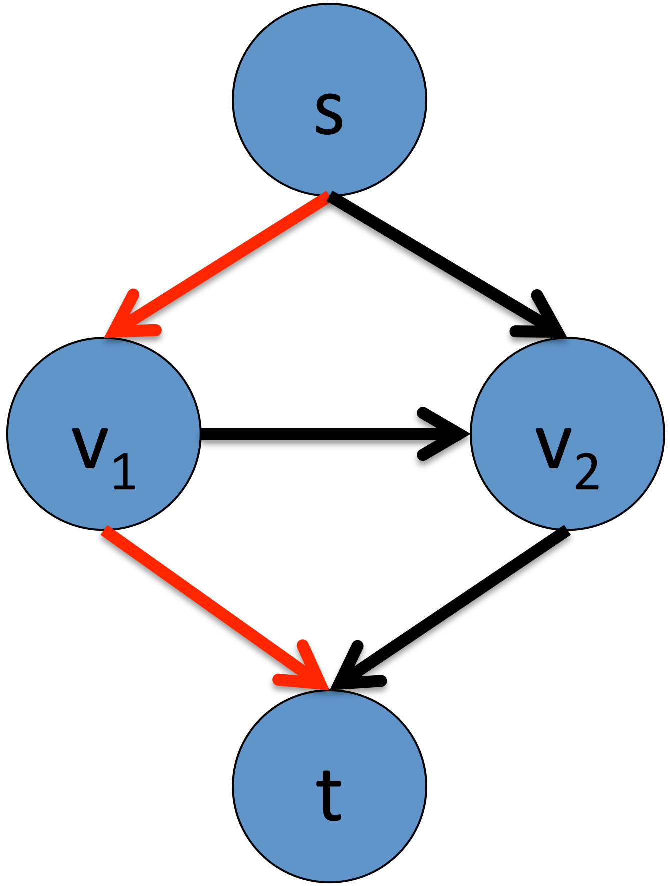




Pass the maximum allowable flow.

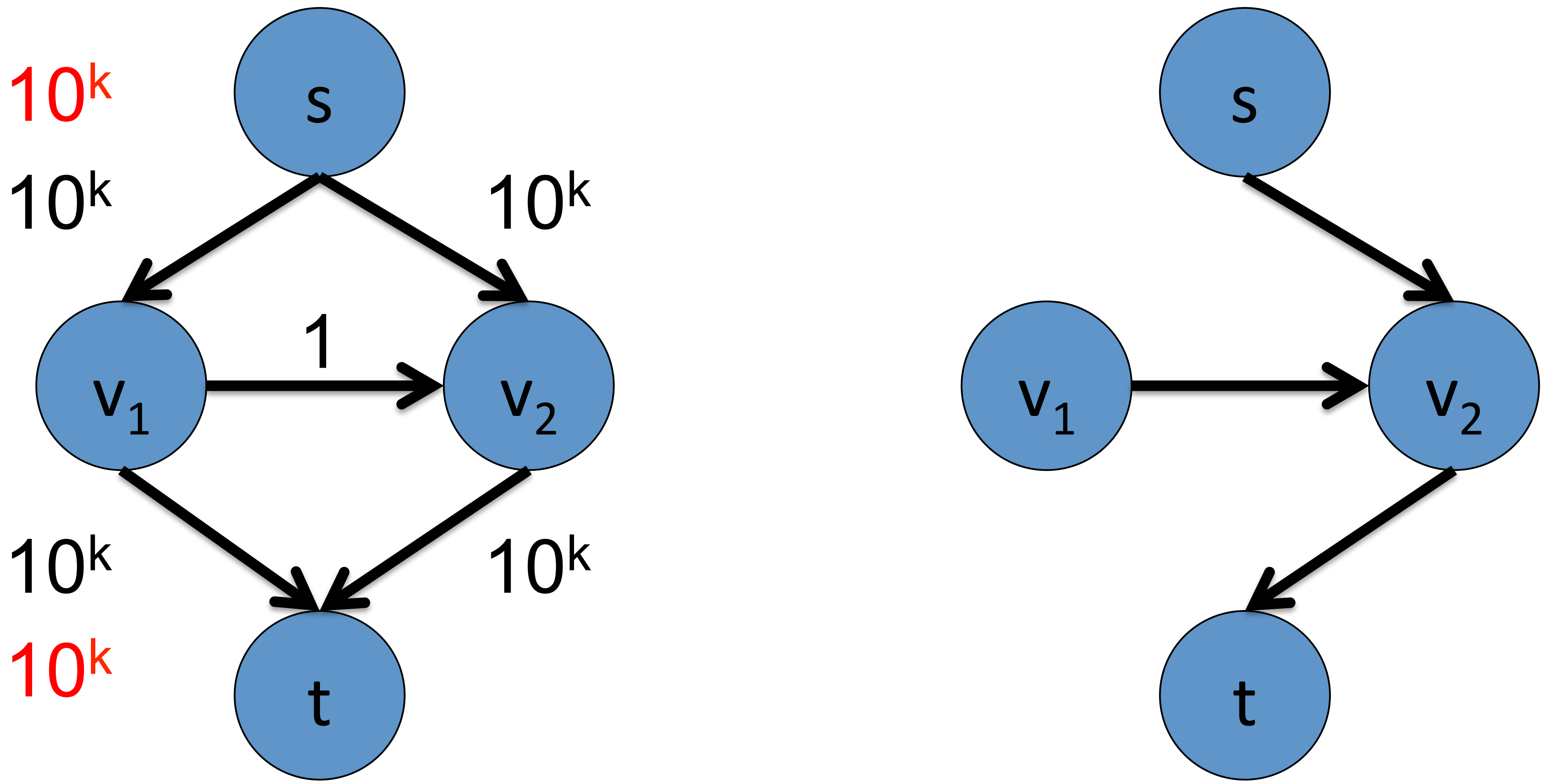
Dinits Algorithm





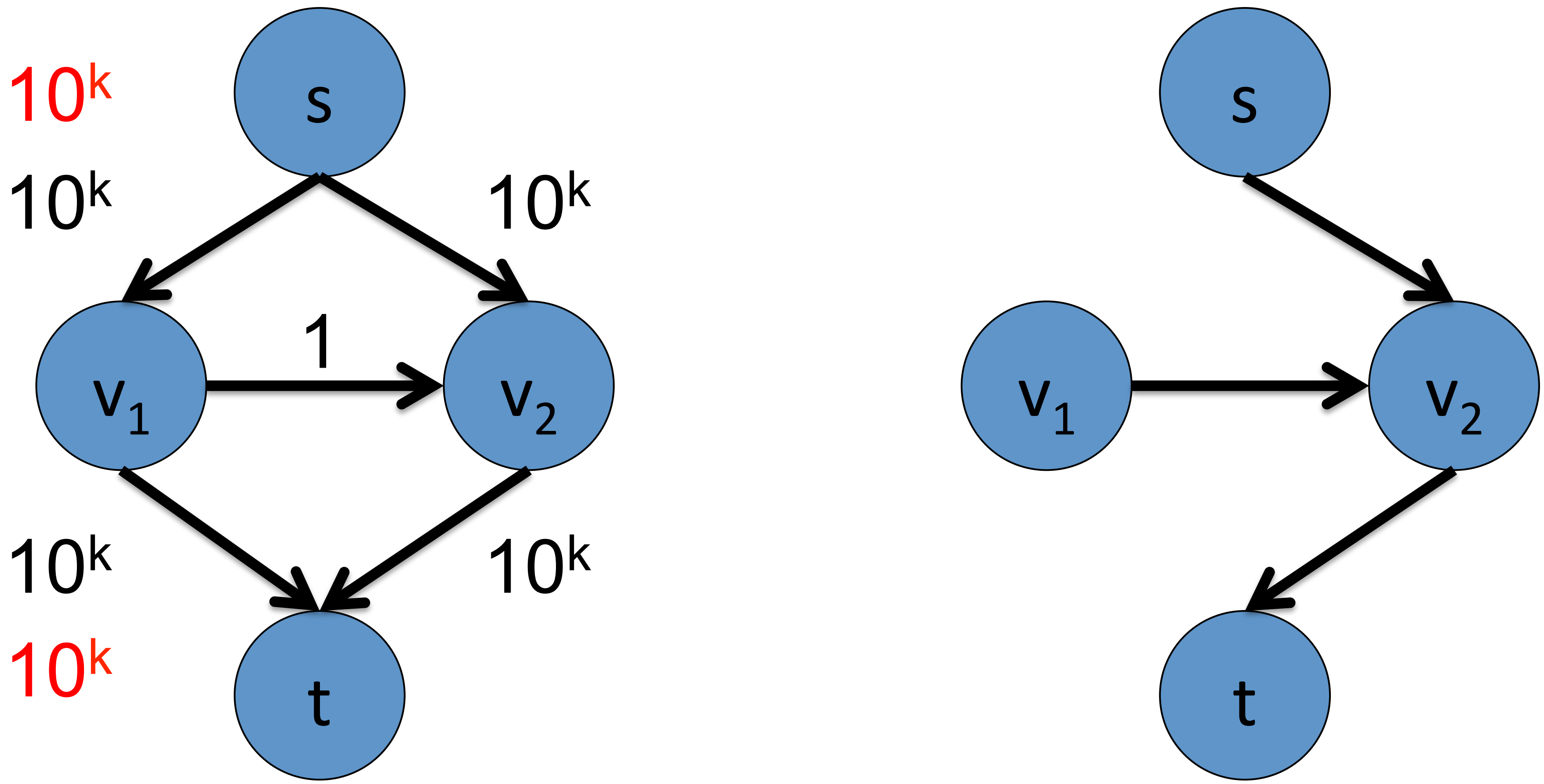
Pass the maximum allowable flow.

Dinits Algorithm



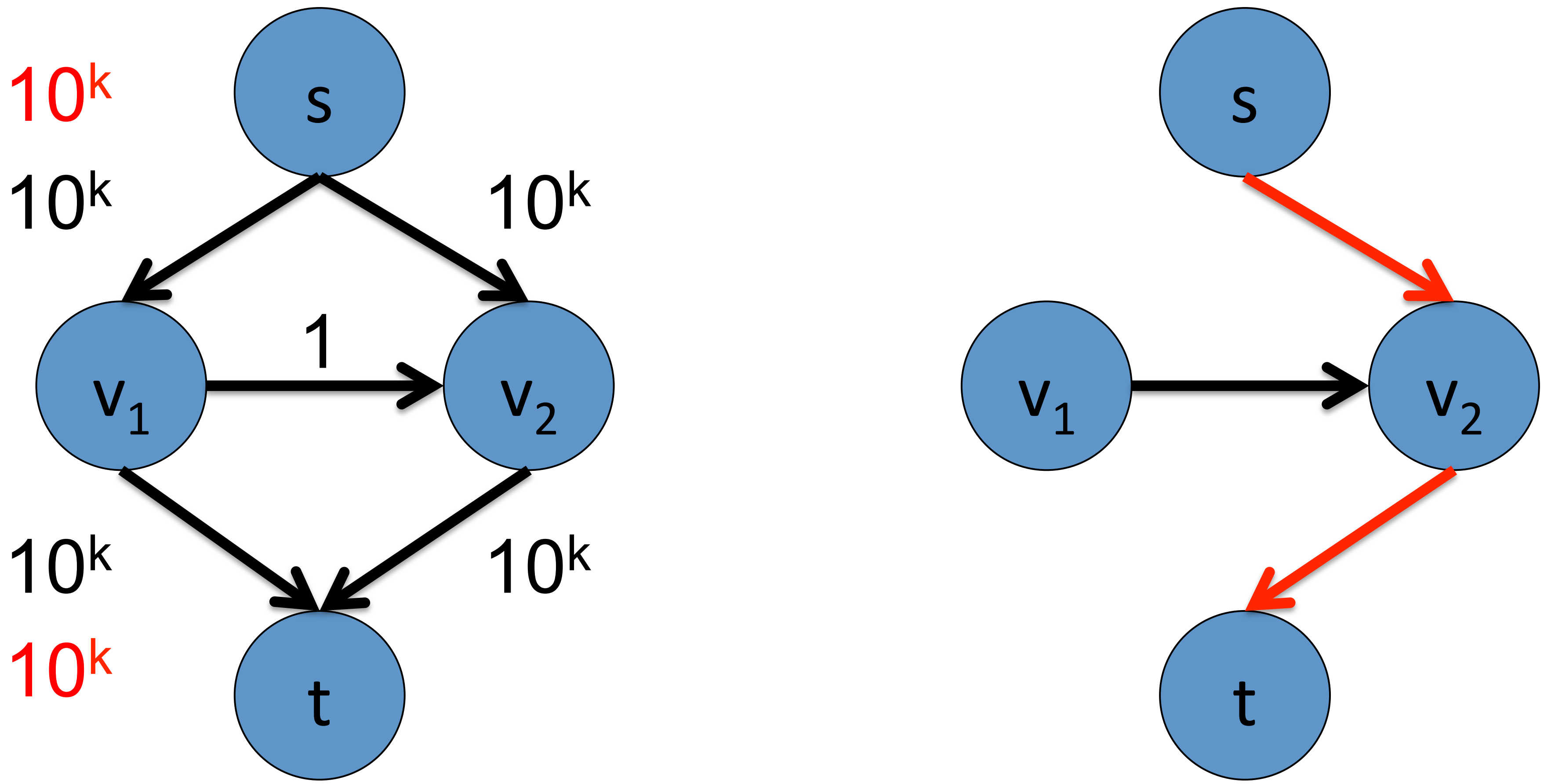
Update the residual graph.

Dinits Algorithm



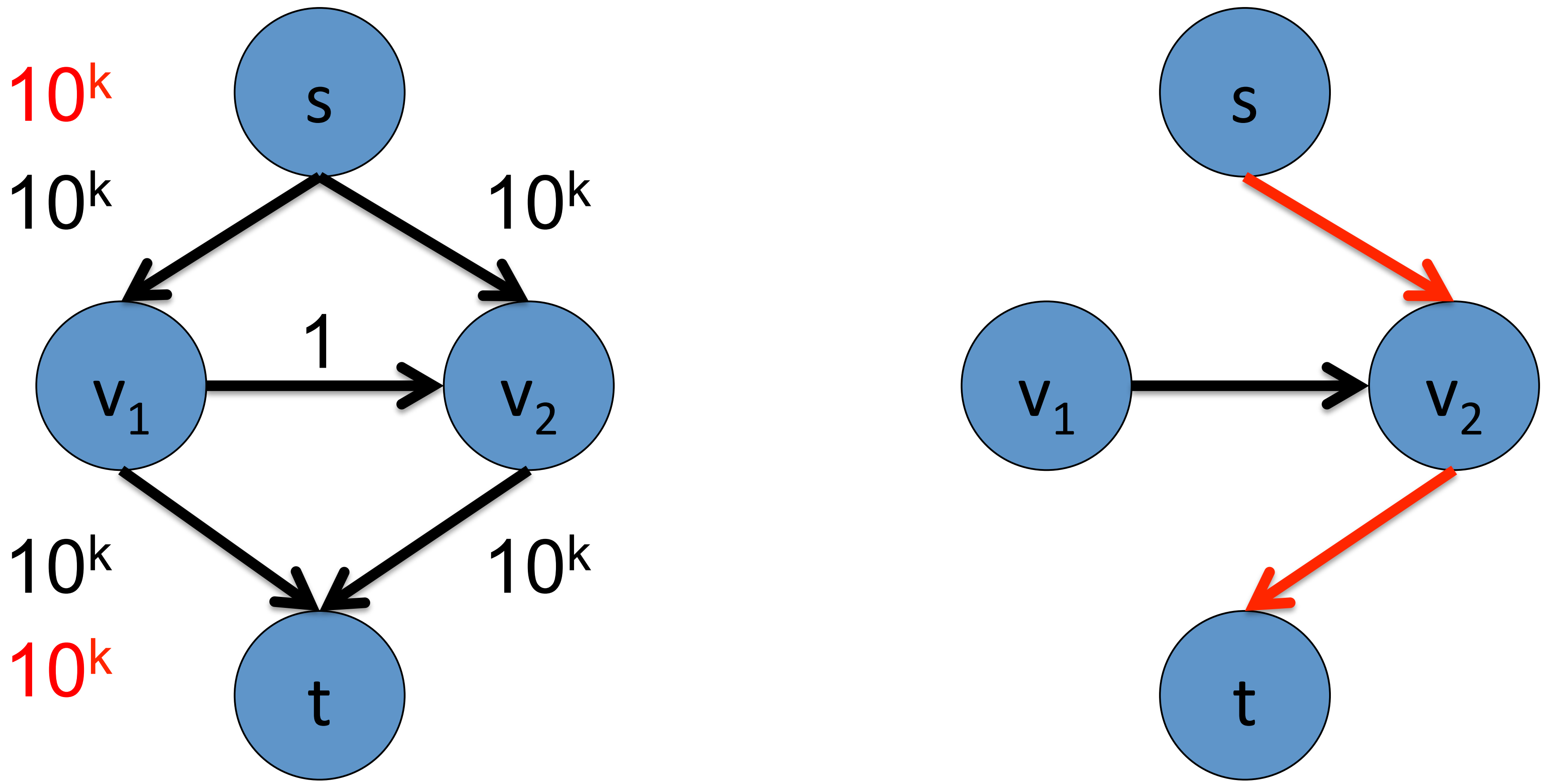
Find the minimum s-t path in the residual graph.

Dinits Algorithm



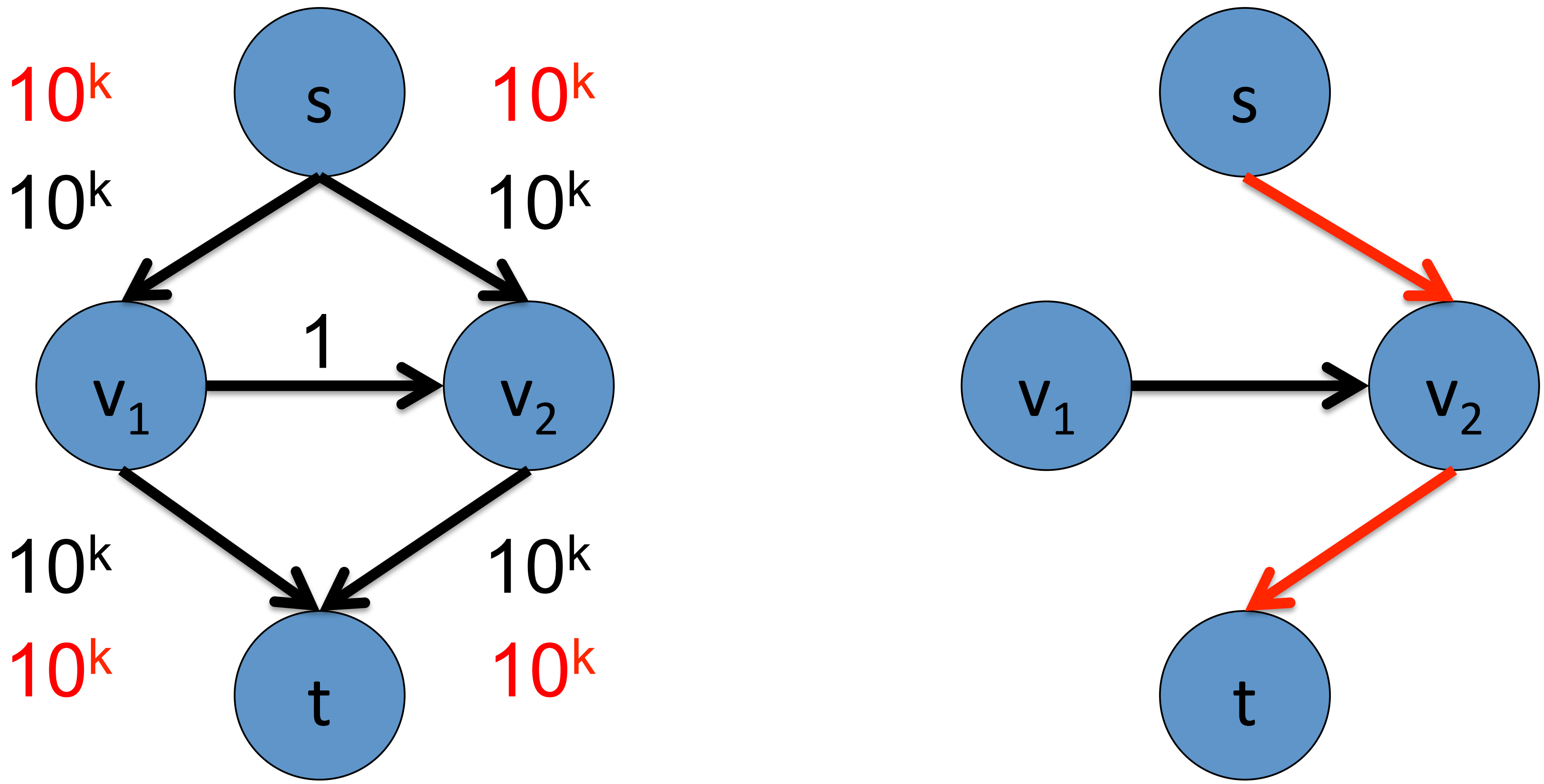
Find the minimum s-t path in the residual graph.

Dinits Algorithm



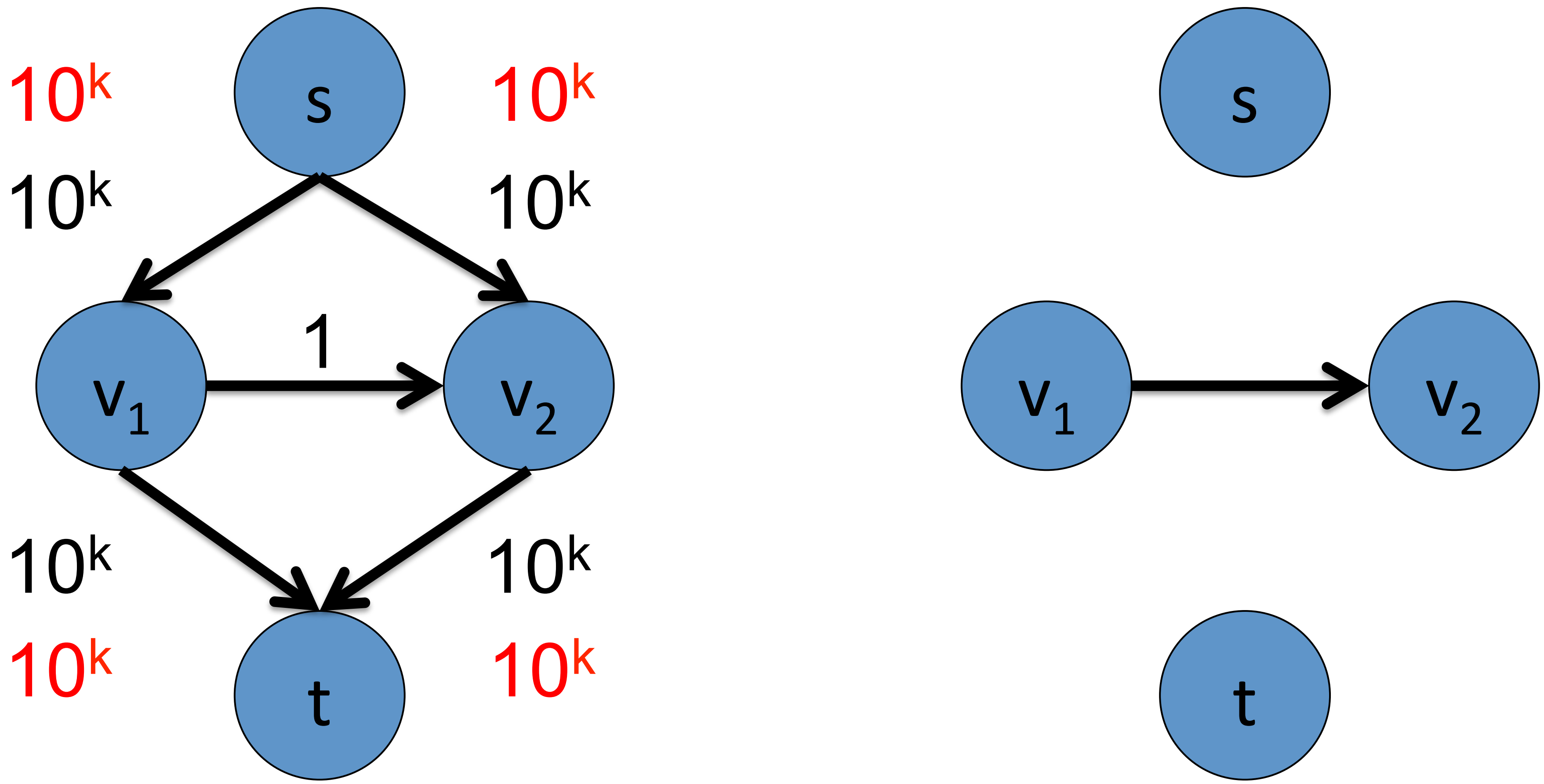
Pass the maximum allowable flow.

Dinits Algorithm



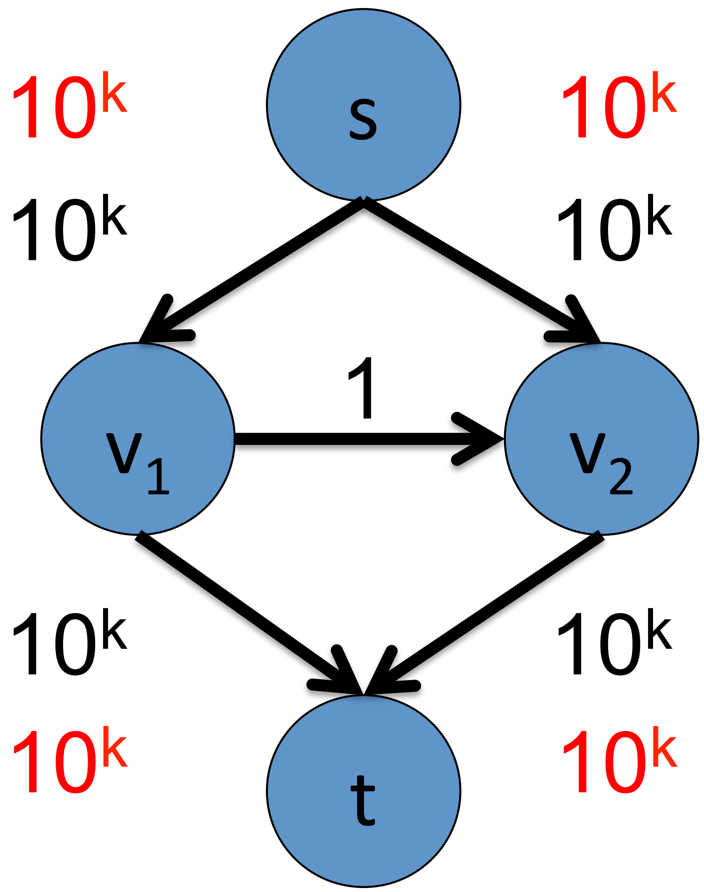
Pass the maximum allowable flow.

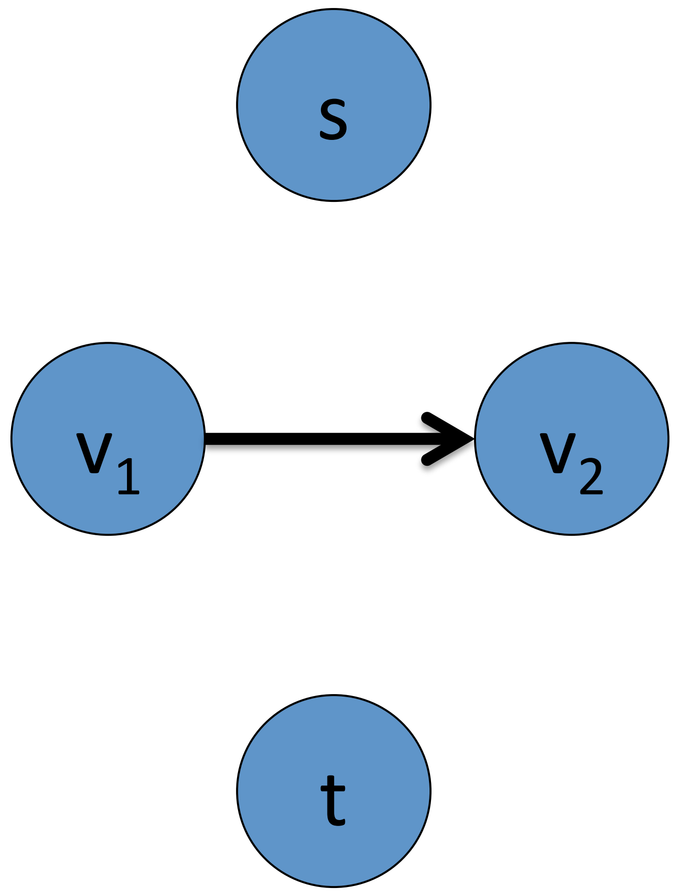
Dinits Algorithm



Update the residual graph.

Dinits Algorithm





No more s-t paths. Stop.

Computational Complexity

Strongly polynomial: $O(m^2n)$ $\quad m = |A|, n = |V|$

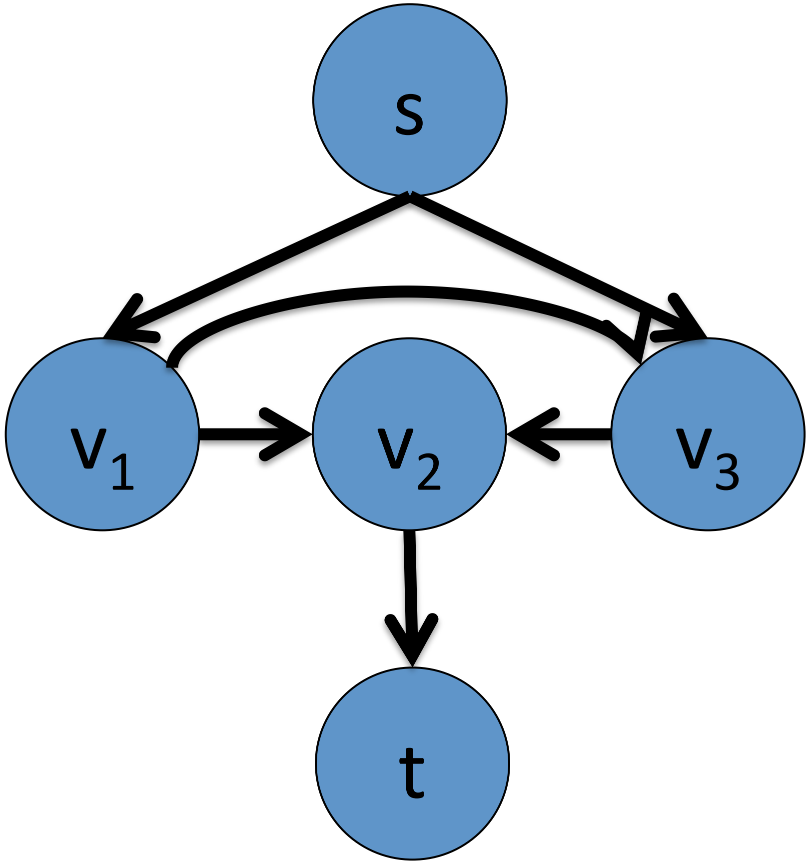
Finding shortest s-t path $\quad O(m)$

Number of iterations $\quad O(mn)$

Proof?

First, a Lemma.

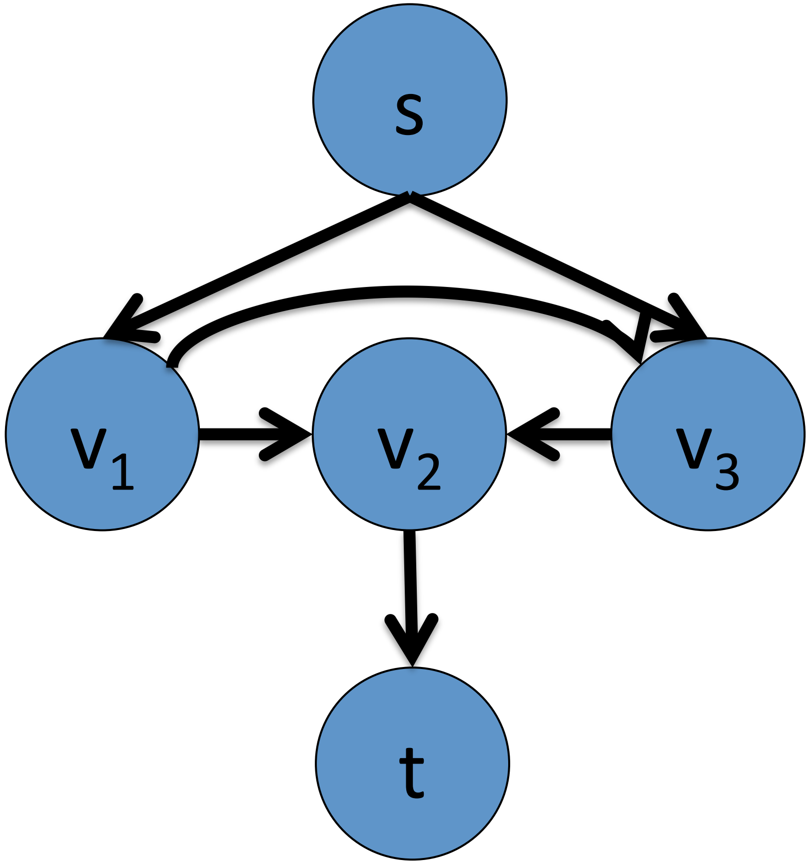
Lemma



$\mu(D) = 3$

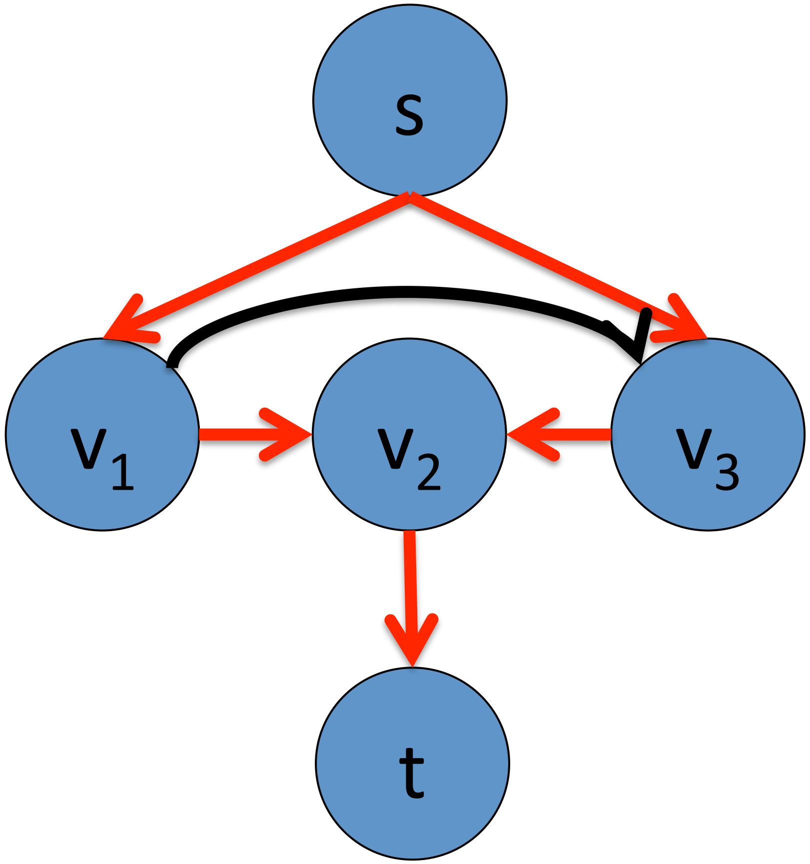
$\mu(D)$ = length of shortest path for D

Lemma



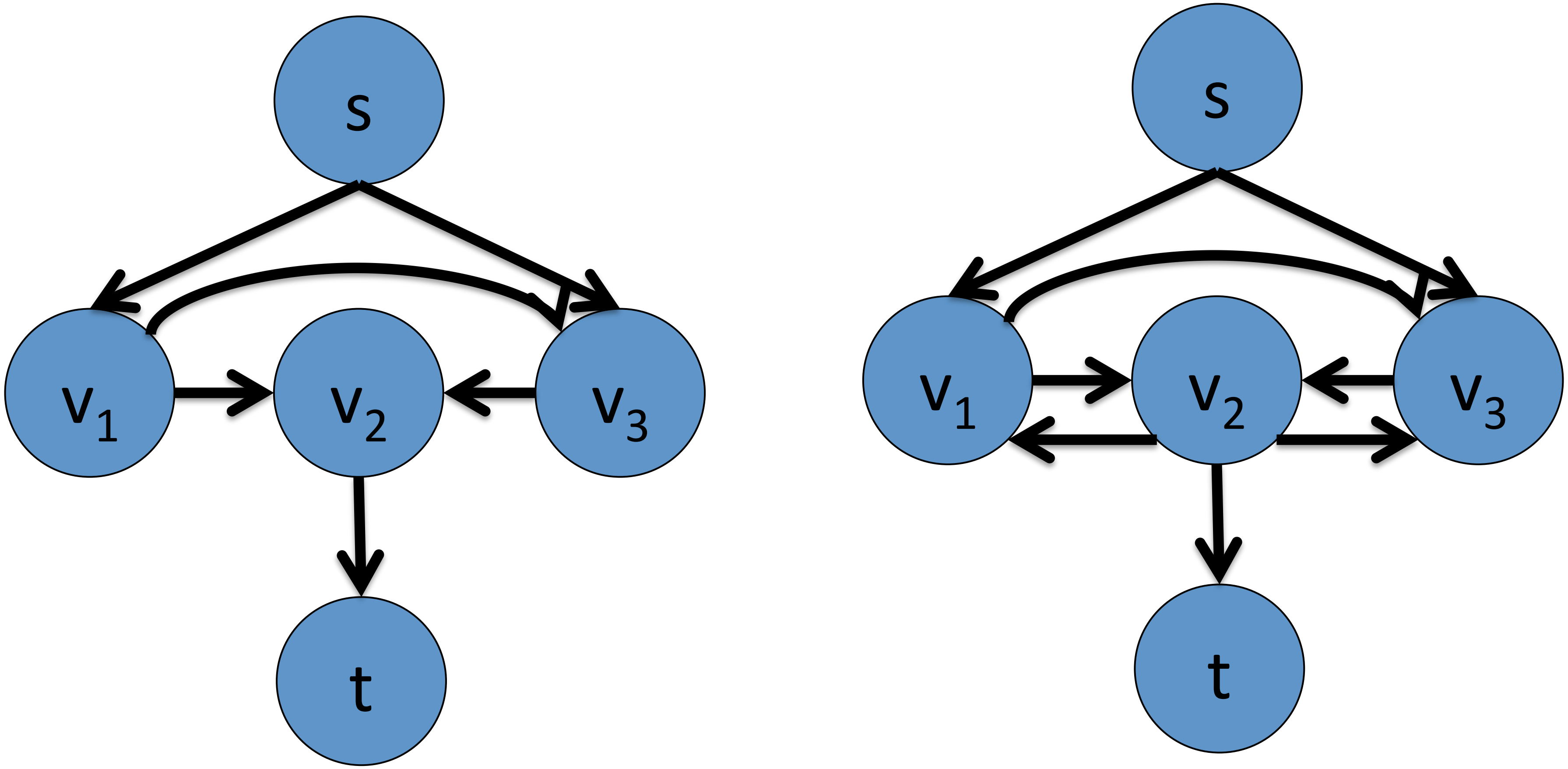
$\alpha(D)$ = arcs present in at least one shortest path

Lemma



α(D) = arcs present in at least one shortest path

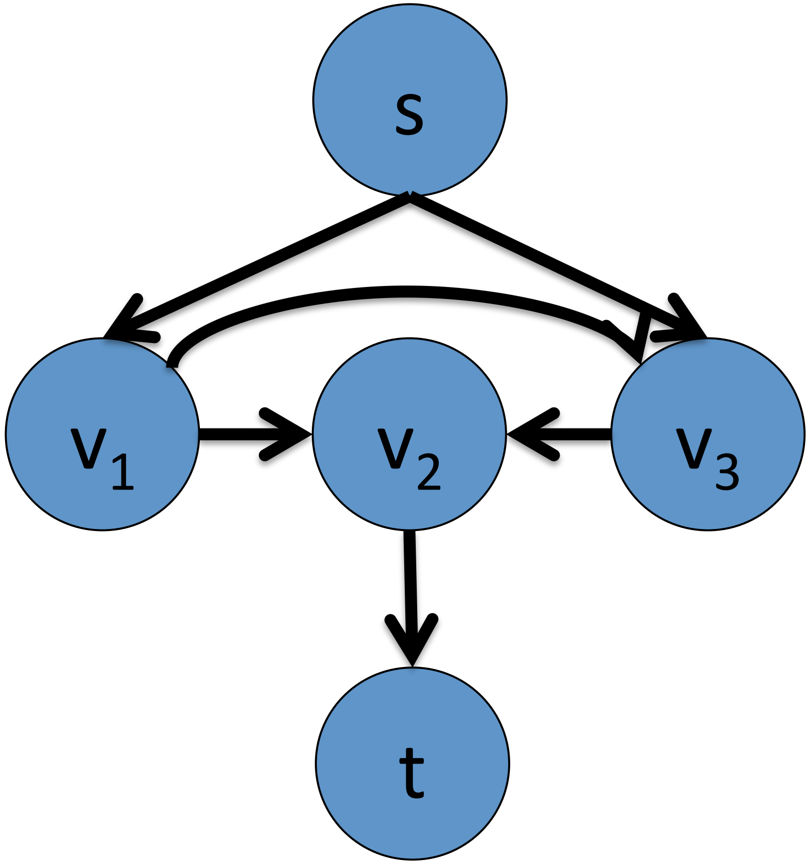
Lemma

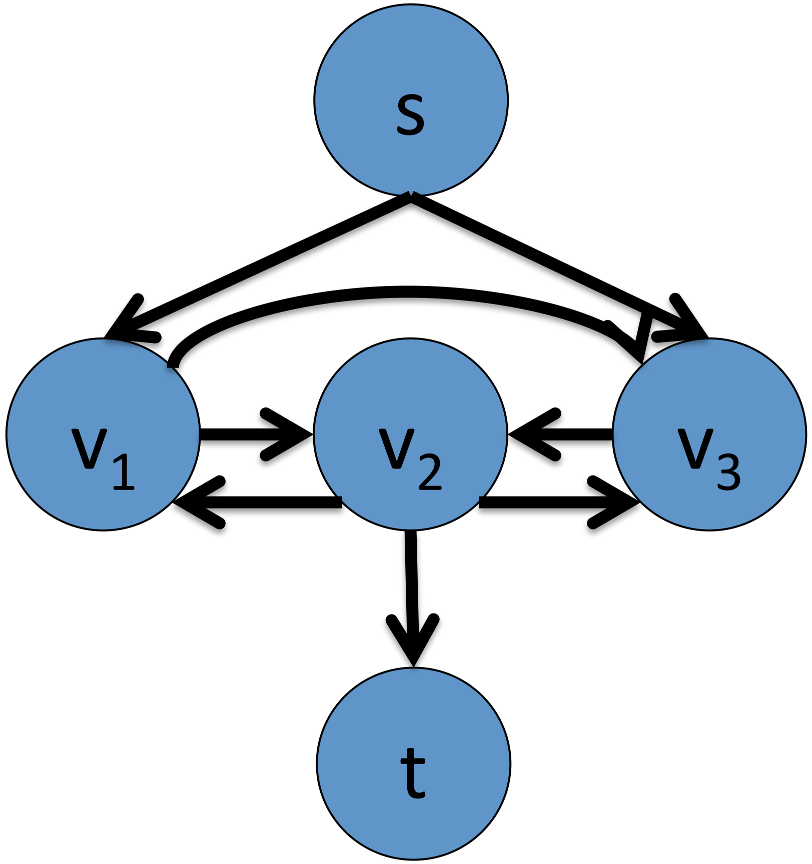


Including arcs to s and from t is not necessary

S(D) = (V, A "union" inverse arcs of α(D))

Lemma





Proof left as exercise !!!

$\mu(D) = \mu(S(D))$

$\alpha(D) = \alpha(S(D))$

Computational Complexity

Strongly polynomial: $O(m^2n)$ $m = |A|, n = |V|$

Finding shortest s-t path $O(m)$

Number of iterations $O(mn)$

Computational Complexity

Current residual graph D_0

Find a shortest path P in D_0

New residual graph D_1

D_1 is a subgraph of $S(D_0)$

$\mu(D_1) \geq \mu(S(D_0)) = \mu(D_0)$

Computational Complexity

At least one arc a in P, $a \in \alpha(D_0)$ and $a \notin \alpha(D_1)$

Specifically, an arc that is saturated in path P

$\mu(D_1) \geq \mu(S(D_0)) = \mu(D_0)$ Assume Equality

$\alpha(D_1)$ "subset of" $\alpha(S(D_0)) = \alpha(D_0)$

Computational Complexity

At least one arc a in P, $a \in \alpha(D_0)$ and $a \notin \alpha(D_1)$

Specifically, an arc that is saturated in path P

$\mu(D_1) \geq \mu(S(D_0)) = \mu(D_0)$ Assume Equality

$\alpha(D_1)$ "strict subset of" $\alpha(S(D_0)) = \alpha(D_0)$

Computational Complexity

At each iteration, either μ increases

Or size of α decreases

Therefore, total iterations = O(mn)

Overall complexity = $O(m^2n)$

Let us take an example

Image segmentation

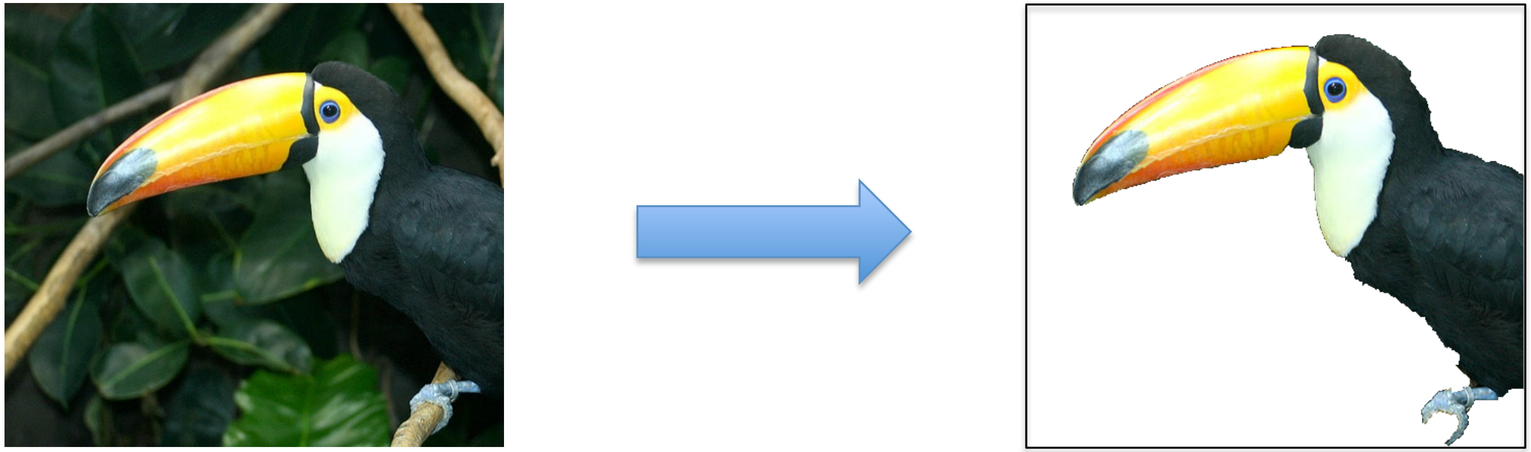


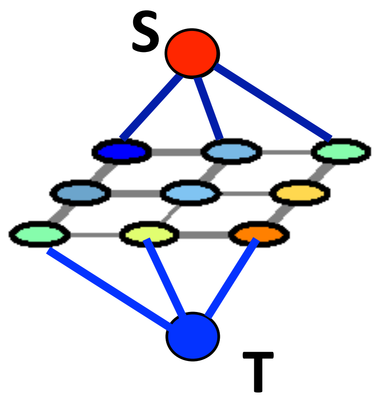
Image Segmentation

Following slides are based on P. Kohli's tutorial

Let us take an example

Image segmentation

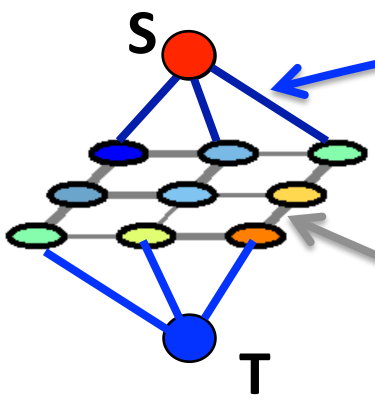
Image



Construct a graph

So how does it work?

Image segmentation



$$E(x) = \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i (1 - x_j)$$

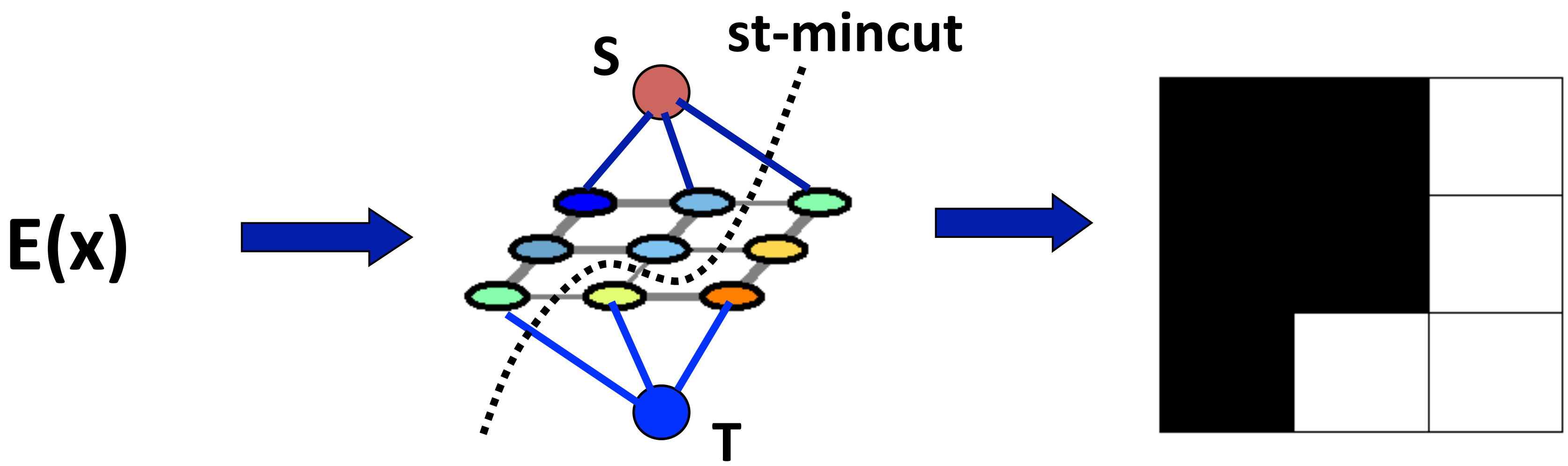
$$E : \{0,1\}^n \rightarrow \mathbb{R}$$

Set the edge weights

How does it work?

Construct a graph such that:

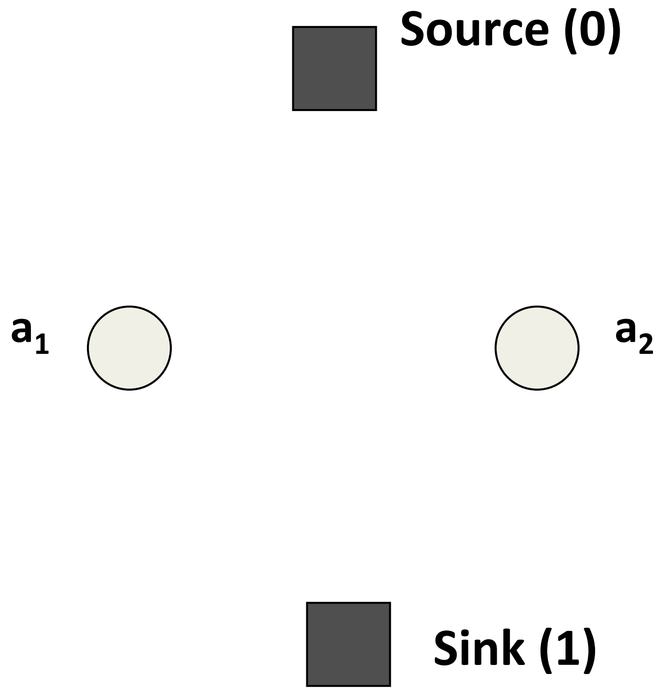
1. Any st-cut corresponds to an assignment of x
2. The cost of the cut is equal to the energy of x : E(x)



Solution

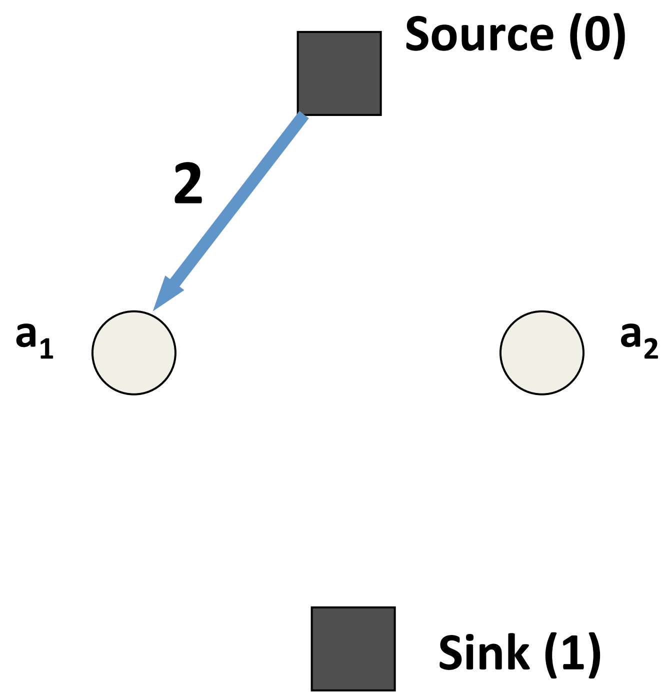
Graph construction

E(a_1,a_2)



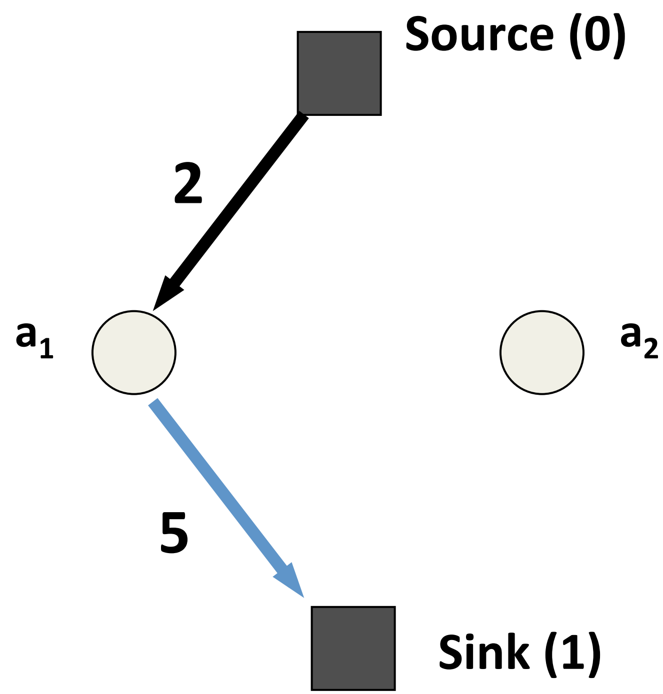
Graph construction

$E(a_1,a_2) = 2a_1$



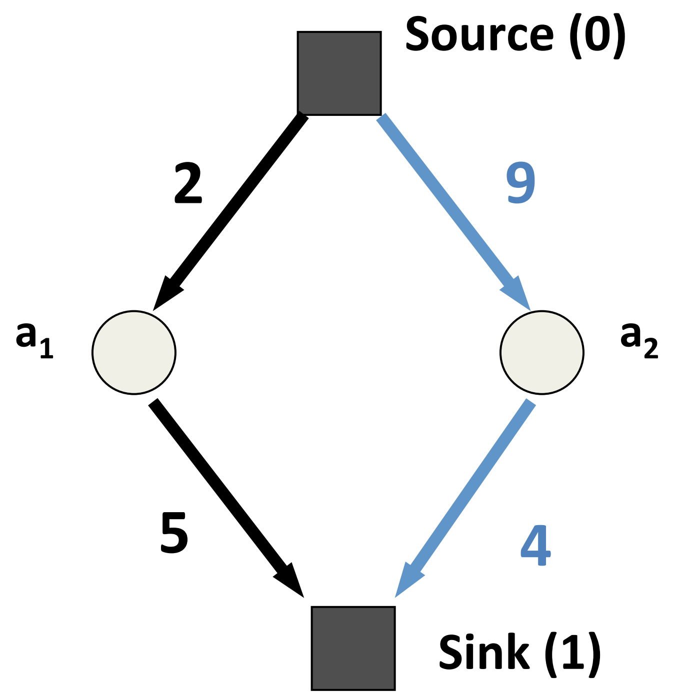
Graph construction

$E(a_1,a_2) = 2a_1 + 5\bar{a}_1$



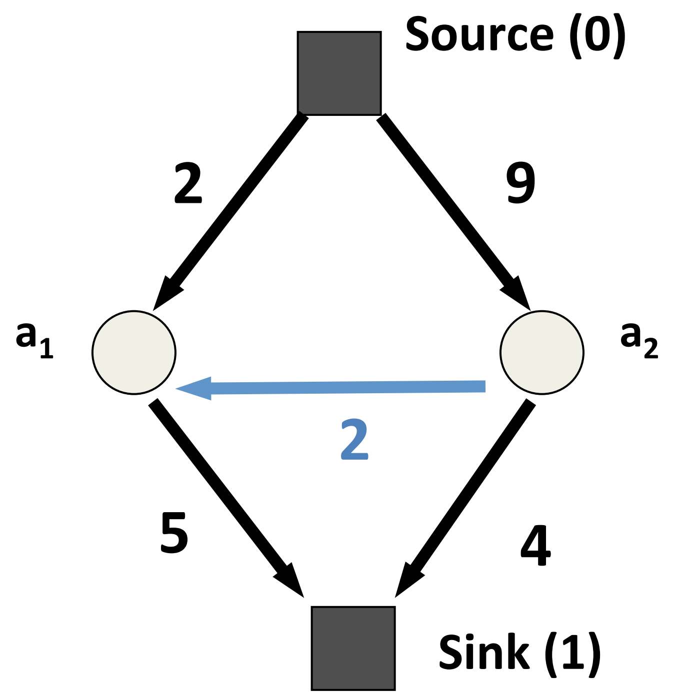
Graph construction

$$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2$$



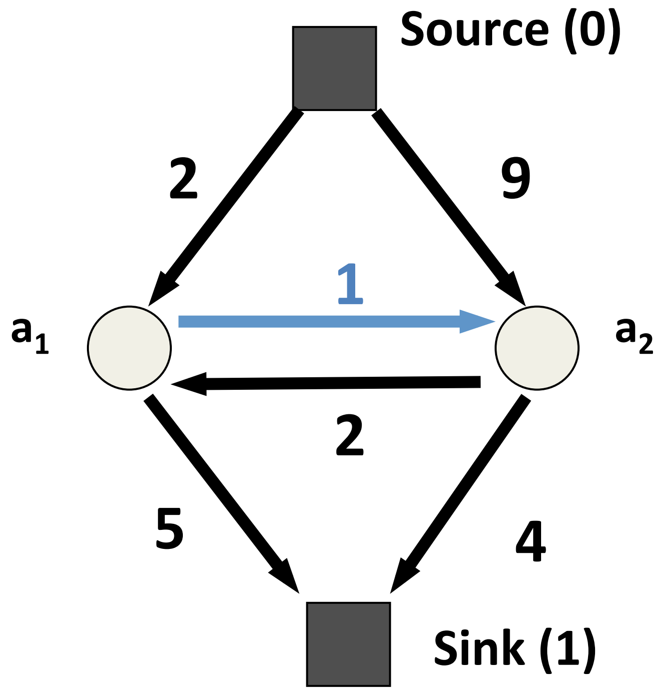
Graph construction

$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2$



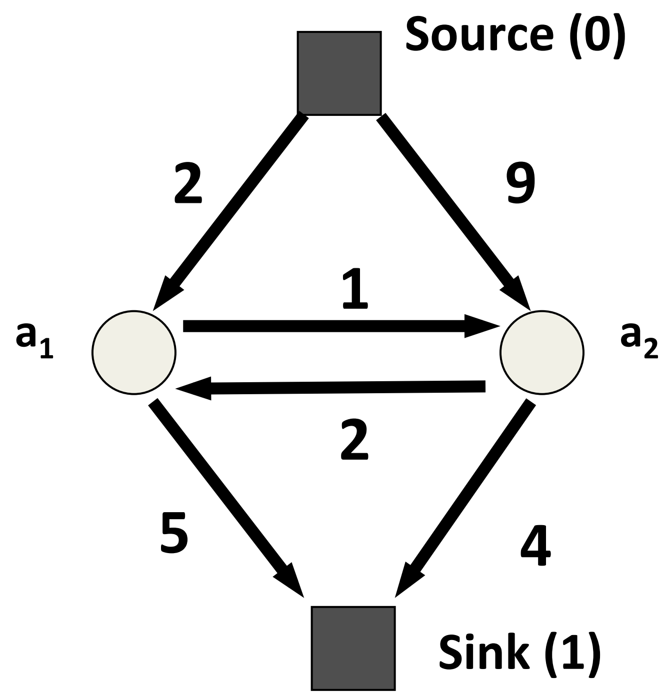
Graph construction

$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$



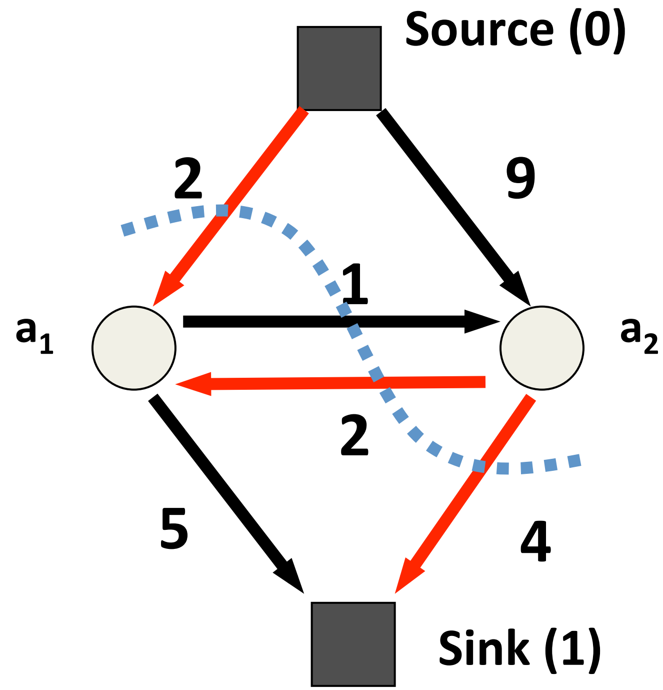
Graph construction

$$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2$$



Graph construction

$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$



st-mincut cost = 8

$a_1 = 1 \quad a_2 = 0$

$E(1,0) = 8$