4.5 Polynomial-time variations and extensions of Ford and Fulkerson algorithm

Assume throughout this section that $\underset{|V|}{n} = O(\underset{|A|}{m})$.

This is not restrictive, because if $m < n-1$, then the graph is disconnected and we can determine the connected component containing the source and focus on that one.

Moreover, let $U := u(A)$ (sum of all capacities)

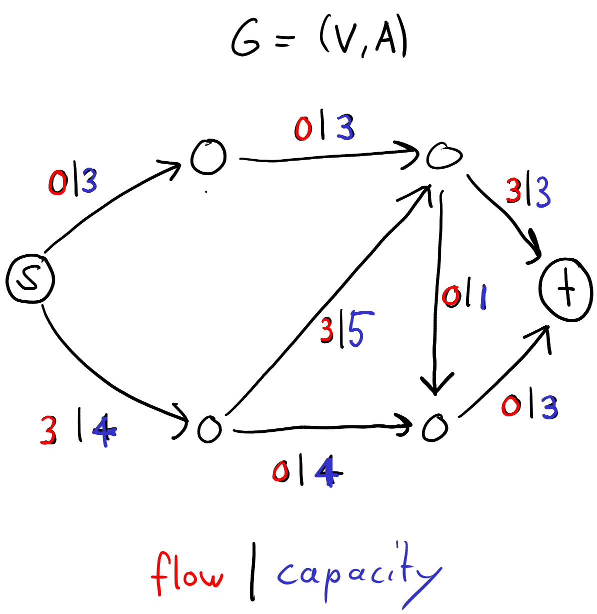
We will discuss 2 efficient maximum flow algorithms:

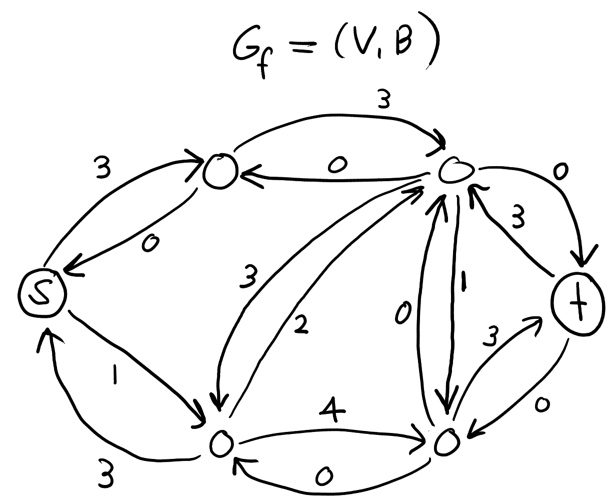
(a) The capacity scaling algorithm

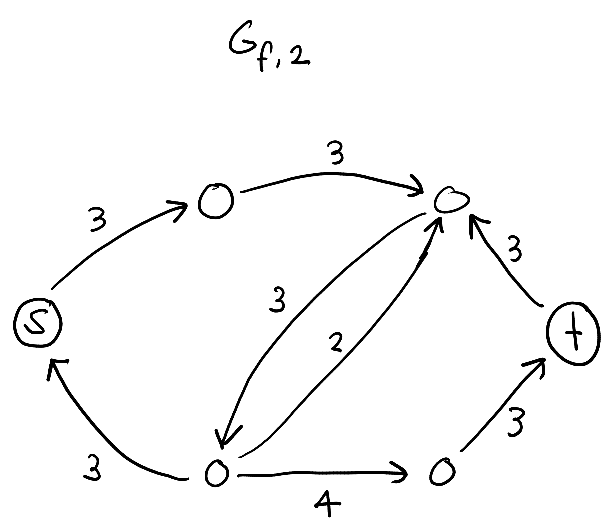
(b) Edmonds-Karp algorithm

4.5.1 Capacity scaling algorithm

> **Definition 4.30:** $G_{f,\Delta}$
>
> Let f be an s-t flow in the directed graph $G = (V, A)$ with capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and let $\Delta \in \mathbb{R}_{\geq 0}$. We denote by $G_{f,\Delta}$ the subgraph of the residual graph $G_f = (V, B)$ containing only the arcs with residual capacity of at least Δ.







Algorithm 6: Capacity scaling algorithm for maximum s-t flows

Input : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum s-t flow f.

```
f(a) = 0  ∀a ∈ A.                         // We start with the zero flow.
Δ = 2^⌊log₂(U)⌋.
while Δ ≥ 1 do                            // These iterations are called phases.
    while ∃f-augmenting path P in G_{f,Δ} do
        Augment f along P and set f to the augmented flow.
    Δ = Δ/2.
return f
```

Theorem 4.31

Algorithm 6 returns a maximum s-t flow.

<u>Proof</u>

Theorem 4.32

Algorithm 6 runs in $O(m^2 \log U)$ time.

Proof

Algorithm 6: Capacity scaling algorithm for maximum s-t flows

Input : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum s-t flow f.

```
f(a) = 0  ∀a ∈ A.                              // We start with the zero flow.
Δ = 2^⌊log_2(U)⌋.
while Δ ≥ 1 do                                  // These iterations are called phases.
    while ∃f-augmenting path P in G_{f,Δ} do
        Augment f along P and set f to the augmented flow.
    Δ = Δ/2.
return f
```

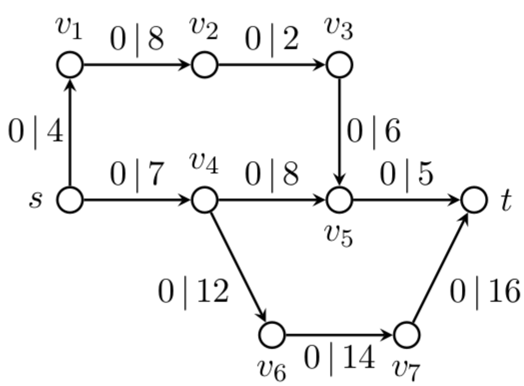
4.5.2 Edmonds-Karp algorithm

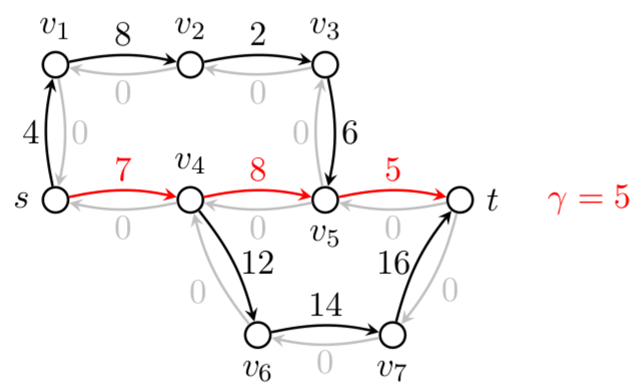
Idea: Augment always on shortest paths.

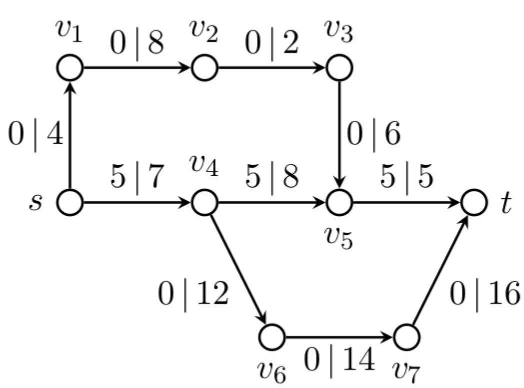
Algorithm 7: Edmonds-Karp algorithm

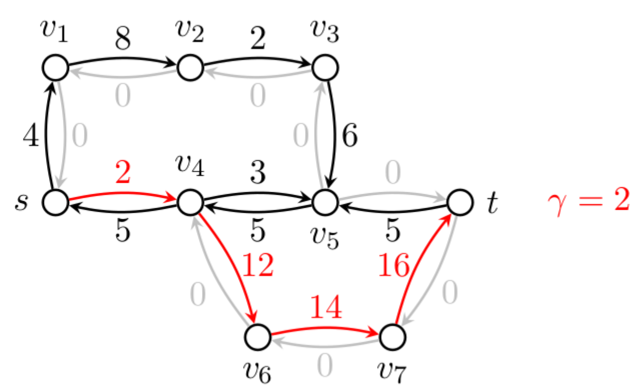
Input : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum s-t flow f.

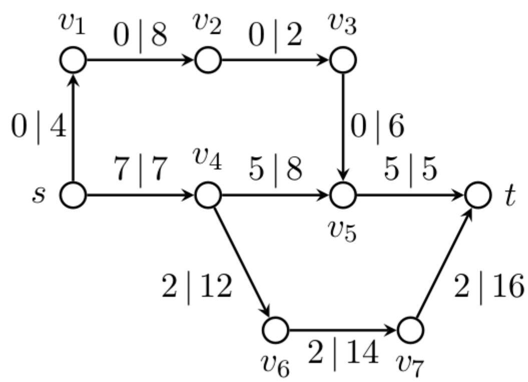
$f(a) = 0 \;\; \forall a \in A$.
while $\exists$ *f-augmenting path in G_f* **do**
  Find an f-augmenting path P in G_f minimizing $|P|$.
  Augment f along P and set f to augmented flow.
return f

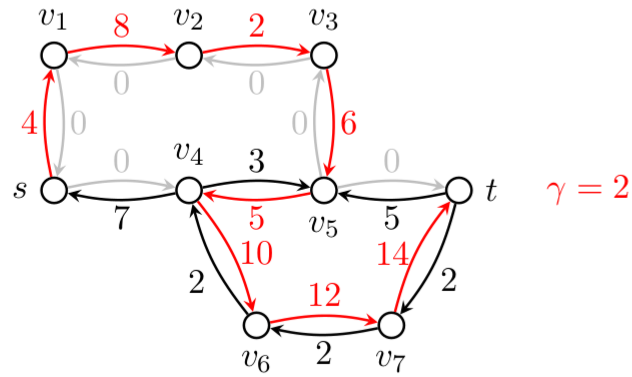


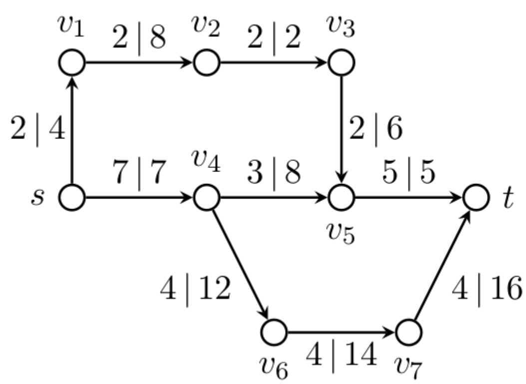


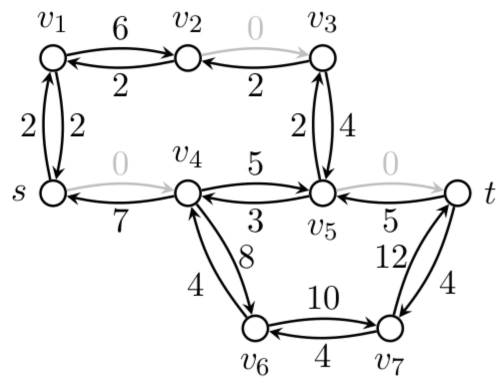





v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
0 | 8
0 | 2
0 | 4
0 | 6
7 | 7
5 | 8
5 | 5
2 | 12
2 | 16
2 | 14


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
$\gamma = 2$


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
2 | 8
2 | 2
2 | 4
2 | 6
7 | 7
3 | 8
5 | 5
4 | 12
4 | 16
4 | 14


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t

Key property: Distances from s and distances to t become larger in residual graphs, when only considering arcs with strictly positive residual capacity.

more formally

Lemma 4.33

Let $G = (V, A)$ be a directed graph with arc capacities $u \colon A \to \mathbb{Z}_{\geq 0}$, and let $s, t \in V$ with $s \neq t$. Moreover, let f_1 be an s-t flow in G, and let f_2 be an s-t flow obtained by augmenting f_1 along a shortest augmenting path P in G_{f_1}. Then,

$$d_{f_1}(s,v) \leq d_{f_2}(s,v) \quad \forall v \in V \ \text{, and}$$
$$d_{f_1}(v,t) \leq d_{f_2}(v,t) \quad \forall v \in V \ ,$$

where $d_f(v,w)$ denotes, for $v, w \in V$ and an s-t flow f, the length (in terms of number of arcs) of a shortest v-w path in G_f that only uses arcs with strictly positive f-residual capacity.

Proof

Theorem 4.34

Algorithm 7 runs in $O(nm^2)$ time.

Proof