# COL 351:
# Analysis and Design of Algorithms

## Lecture 32

# Max-Flow: Edmonds-Karp-Algorithm-1

Edmonds-Karp-Algorithm-1$(G, s, t)$:

1. Initialise $f = 0$
2. **While**($\exists\ s \to t$ path in $G_f$):
   - 2.1 Let $P$ be an $s \to t$ **shortest-path** in $G_f$
   - 2.2 Let $c_{min} = \min\{c(e) \mid e \in P\}$
   - 2.3 **For each** $(x, y) \in P$ :
     - If $(x, y)$ is forward edge : $f(x, y) = f(x, y) + c_{min}$
     - If $(x, y)$ is backward edge : $f(x, y) = f(x, y) - c_{min}$
3. Return $f$.

**Claim:**

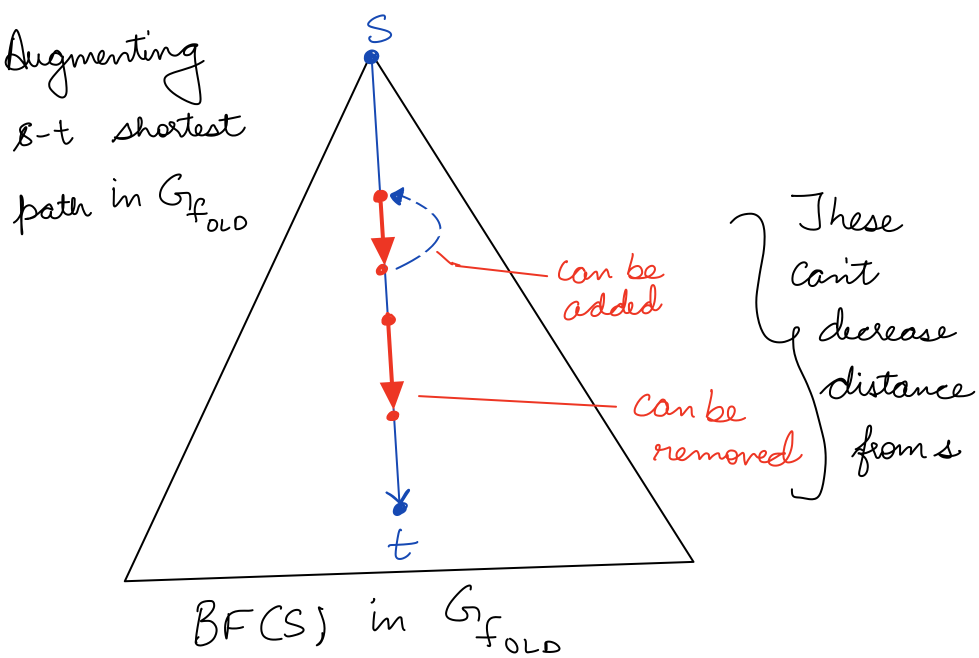
Number of iterations is $O(mn)$

**Time Complexity:** $O(mn(m+n))$

Note: $G_f$ is unweighted.

# Observations

**Claim 1:** The distances of vertices from $s$ in $G_f$ can only increase with time.




When we move from $G_{f_{old}}$ to $G_{f_{new}}$ distance of vertices from 's' CANNOT decrease.

# How many iterations?

**Claim 2:** Total number of iterations is

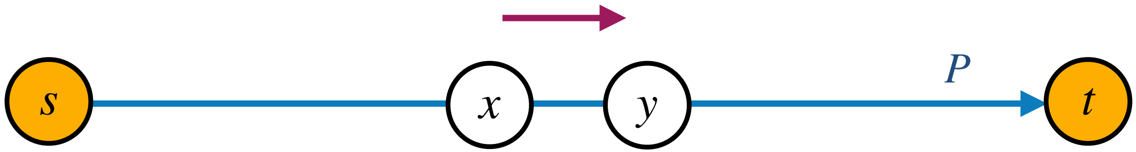
$$O(\text{Total number of times edges disappear from } G_f).$$

Proof :

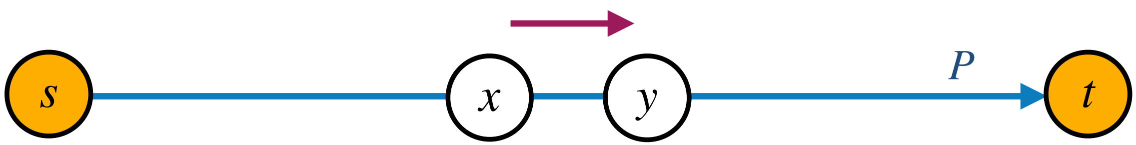
In each iteration at least ONE edge disappears from $G_f$

# What causes disappearance of an edge from $G_f$

$(x, y)$ is
Forward edge
in $G_f$

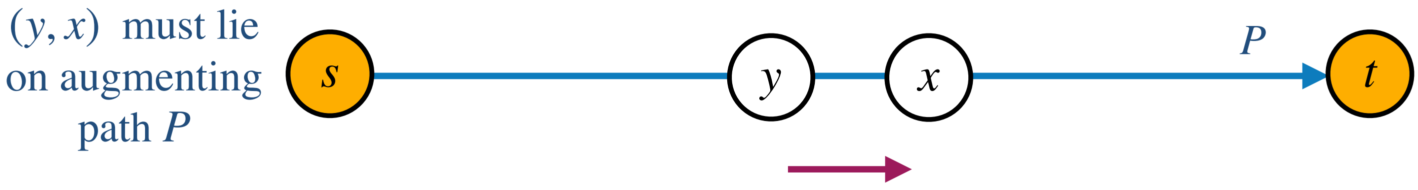



$(x, y)$ is
Backward edge
in $G_f$




If edge $(x,y)$ disappears from $G_f$ then irrespective of whether it is FWD/BACK edge, the edge $(x,y)$ lies on augmenting path.

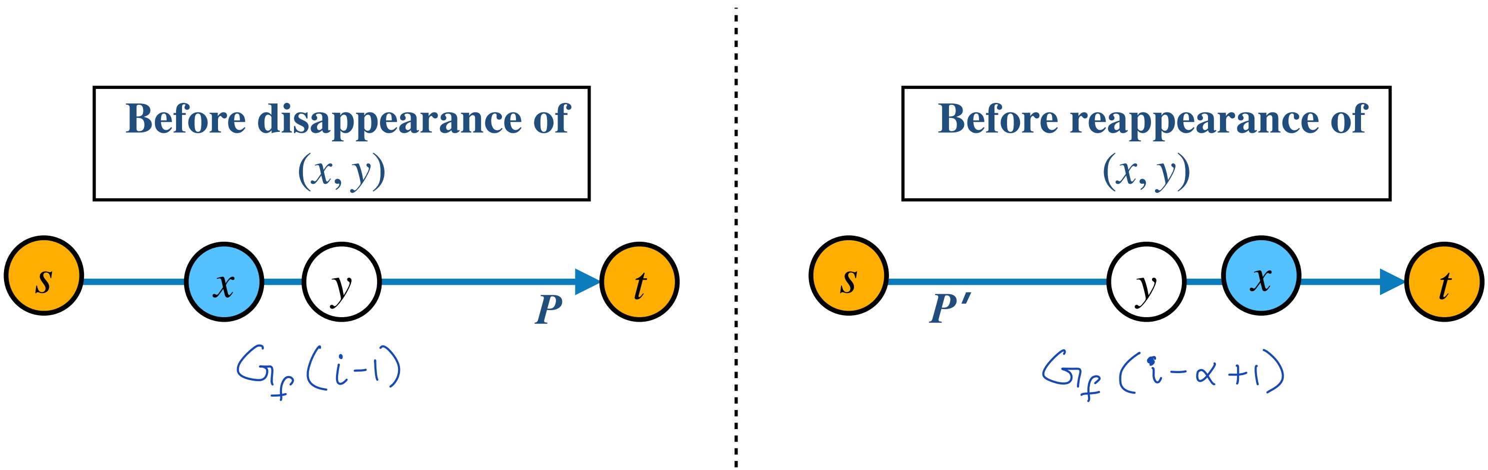
# What causes reappearance of an edge in $G_f$




If edge $(x,y)$ reappears in $G_f$ then the last augmenting path must have contained edge $(y,x)$.

# How many times can an edge disappear/re-appear?

**Claim 3:** An edge $(x, y)$ can disappear/re-appear in $G_f$ at most $O(n)$ time.




Suppose edge $(x,y)$ disappears in $G_f(i-1)$ and reappears in $G_f(i+\alpha)$.

Then,

$$d(s,y,i-1) = d(s,x,i-1) + 1 \quad \text{and} \quad d(s,y,i+\alpha-1) = d(s,x,i+\alpha-1) - 1.$$

Further, $d(s,y,i+\alpha-1) \geq d(s,y,i-1)$ due to Claim 1.

Thus, $d(s,x,i+\alpha-1) \geq d(s,x,i-1) + 2$

# Max-Flow: Edmonds-Karp-Algorithm-2

Edmonds-Karp-Algorithm-2$(G, s, t)$:

1. Initialise $f = 0$
2. **While**($\exists\ s \to t$ path in $G_f$):

    2.1 Let $P$ be an $s \to t$ **path of maximum capacity** in $G_f$

    2.2 Let $c_{min} = \min\{c(e) \mid e \in P\}$

    2.3 **For each** $(x, y) \in P$ :

    If $(x, y)$ is forward edge : $f(x, y) = f(x, y) + c_{min}$

    If $(x, y)$ is backward edge : $f(x, y) = f(x, y) - c_{min}$
3. Return $f$.

**Claim:**

Number of iterations is $O(m \log_e F)$

Overall time complexity :

$$\underbrace{O(m \log_e F)}_{\text{No of iterations}} \times \underbrace{O(m \log n)}_{\text{Time to find max capacity path}}$$

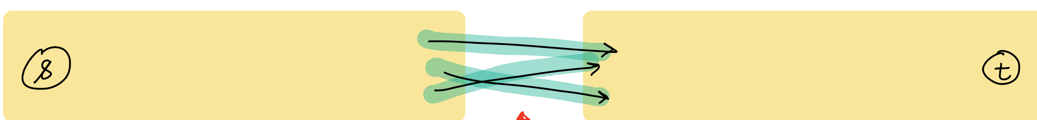
# Observations

**Claim 1:** If in graph $G$, the value of $(s, t)$-max-flow is $F$ then there must exists an $(s, t)$-path in $G$ of capacity at least $(F/m)$.

Proof:

Discard all edges of capacity $\leq F/m$, & let $H$ be the new graph.

Assume on contrary there is no $(s,t)$-path in $H$.




Let $X$ = vertices reachable from $s$ in $H$.

capacity $\leq F/m$

$Y$ = vertices not reachable from $s$ in $H$.

$(X,Y)$ is an $(s,t)$ cut in $G$ of capacity $\leq \frac{F}{m} \cdot m \leq F$

This leads to contradiction $\Rightarrow$ There exists an $(s,t)$ path in $H$.

# Observations

Fact: For $k \geq 1$, $0 \leq \left(1 - \frac{1}{k}\right)^k \leq \frac{1}{e}$

**Claim 2:** The smallest $x$ for which $F\left(1 - \dfrac{1}{m}\right)^x$ is smaller than 1 is at-most $(m \log_e F)$.

Proof:

After $x$ iterations, max-flow $\leq F(1 - 1/m)^x$

We need to find smallest $x$ s.t. $F\left(1 - \frac{1}{m}\right)^x < 1$

Put $x = m \log_e F$.

$$F\left(1 - \frac{1}{m}\right)^{m \cdot \log_e F} \leq F\left(\frac{1}{e}\right)^{\log_e F} = \frac{F}{F} = 1$$

# Observations

**Claim 3:** We can find an augmenting path of maximum capacity in residual graph $G_f$ in $O(m \log n)$ time.

If $x_1, \ldots, x_t$ are in-neighbors of $y$, then

$$\mathrm{MaxCap}(s, y) = \max_{1 \leqslant i \leqslant t} \left( \min \left( \mathrm{MaxCap}(s, x_i), c(x_i, y) \right) \right)$$

Proof : H. W.