# Ford-Fulkerson Algorithm for Maximum Flow

1. Assign an initial flow $f_{ij}$ (for instance, $f_{ij}=0$) for all edges

2.Label s by Ø. Mark the other vertices "unlabeled."

3. Find a labeled vertex i that has not yet been scanned . Scan i as follows

For every unlabeled adjacent vertex j, (**a** or **b** or **c**)

**a)** if $C_{ij} > f_{ij}$ and $f_{ij} \geq 0$

compute $\Delta_{ij} = C_{ij} - f_{ij}$ and $\Delta_j$ where

$$\Delta_j = \begin{cases} \Delta ij & \text{if } i = 1 \\ \min(\Delta i,\ \Delta ij) & \text{if } i > 1 \end{cases}$$

Label j with a forward label $(i^+, f_{ij})$

**b)** if $C_{ij} > |f_{ij}|$ and $f_{ij} < 0$ (opposite direction)

$\Delta_j = \min(\Delta_j, |f_{ij}|)$

Label j with a backward label $(i^-, \Delta_j)$

**c)** if $C_{ij} = f_{ij}$ No operation.

If no unlabeled j exists STOP.

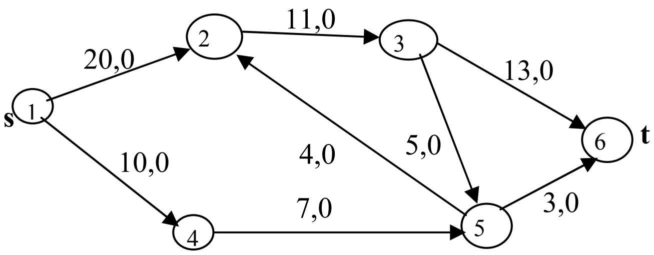
**4)** Repeat step 3 until t is reached.
[This gives a flow augmenting path P: s —>• t]
If it is impossible to reach t then STOP.

**5)**. Backtrack the path P, using the labels.

**6)**Using P, augment the existing flow by $\Delta_t$,
Set $f = f + \Delta_t$,.

**7)** Remove all labels from vertices 2, … , n.
Go to Step 3.

**Example:** Find the maximum flow from **s** to **t** in the following graph.




**Solution 1)** $C_{12}=20$, $C_{23}=11$, $C_{36}=13$, $C_{35}=5$, $C_{14}=10$, $C_{45}=7$, $C_{52}=4$, $C_{56}=3$,
$f_{12}=f_{23}=f_{36}=f_{35}=f_{14}=f_{45}=f_{52}=f_{56}=0$,

**2)** vertex 1 (s) is **labeled Ø**, 2,3,4,5,6 are **unlabeled**

**3) Scan 1. i**=1 ,Adjacent labels 2 and 4. [ **j**=2 and **j**=4]

$C_{12}=20$. $f_{12}=0$. (perform **a**)

For vertex j=2

$\Delta_{12} = C_{12} - f_{12} = 20 - 0 = 20$

$\Delta_2 = \Delta_{12} = 20$.

**L2** = $\{1^+, 20\}$

For vertex j=4

$C_{14}=20$. $f_{14}=0$. (perform **a**)

$\Delta_{14} = C_{14} - f_{14} = 10 - 0 = 10$

$\Delta_4 = \Delta_{14} = 10$.

**L4** = $\{1^+, 10\}$

**Scan 2. i**=2 ,Adjacent labels 1, 3 and 5. [ **j**=3 and **j**=5]
(j=1 is already labelled)

For vertex j=3

$C_{23}=11$. $f_{23}=0$.

$\Delta_{23} = C_{23} - f_{23} = 11 - 0 = 11$

$\Delta_3 = \min(\Delta_2, \Delta_{23}) = \min(20, 11) = 11$

**L3** = $\{2^+, 11\}$

For vertex j=5 , $f_{25} < 0$ **(perform b)**

$\Delta_5 = \min(\Delta_2 - |f_{25}|) = \min(20,0) = 0$

**L5** = $\{2^-, 0\}$

**Scan 3. i**=3 ,Adjacent labels 2, 5 and 6. [ **j**=6 ]
(j=2 and j=5 are already labelled)

$C_{36}=13$. $f_{36}=0$.

$\Delta_{36} = C_{36} - f_{36} = 13 - 0 = 13$

$\Delta_6 = \min(\Delta_3, \Delta_{36}) = \min(11, 13) = 11$

**L6** = $\{3^+, 11\}$

Since **vertex 6** **is t** $\boxed{\Delta_t=11}$

---

**Now all vertices are all labeled**

Find the path

**L6** = $\{3^+, 11\}$ → **L3** = $\{2^+, 11\}$ → **L2** = $\{1^+, 20\}$ → **L1**

Thus one augmenting path is 1-2-3-6

Add $\boldsymbol{\Delta_t}=11$ to this path

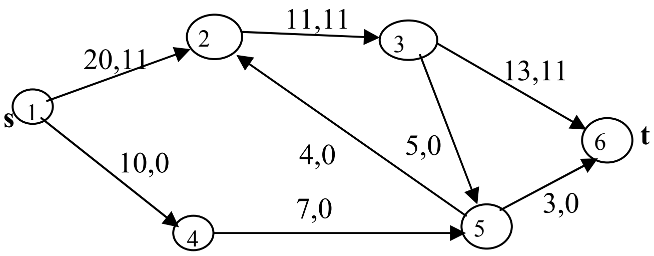
$f_{12(new)} = f_{12(old)} + \boldsymbol{\Delta_t}$

$f_{12} = 0 + 11$

$f_{23} = 0 + 11 = 11$

$f_{36} = 0 + 11 = 11$

**Remove all the labels. Start scanning**




**1)** $C_{12}=20$, $C_{23}=11$, $C_{36}=13$, $C_{35}=5$, $C_{14}=10$, $C_{45}=7$, $C_{52}=4$, $C_{56}=3$,

$f_{12}=f_{23}=f_{36}=11$, $f_{35}=f_{14}=f_{45}=f_{52}=f_{56}=0$,

**2)** vertex 1 (s) is **labeled Ø**, 2,3,4,5,6 are **unlabeled**

**3) Scan 1. i**=1 ,Adjacent labels 2 and 4. [ **j**=2 and **j**=4]

$C_{12}=20$. $f_{12}=11$. (perform **a**)

For vertex j=2

$\Delta_{12} = C_{12} - f_{12} = 20 - 11=9$

$\Delta_2 = \Delta_{12} =9$.

**L2** = {1$^+$, 9}

For vertex j=4

$C_{14}=10$. $f_{14}=0$.

$\Delta_{14} = C_{14} - f_{14} = 10 - 0=10$

$\Delta_4 = \Delta_{14} =10$.

**L4** = {1$^+$, 10}

**Scan 2. i**=2 , Adjacent labels 1, 3 and 5. [ **j**=3 and **j**=5] (j=1 is already labelled)

For vertex j=3,

$C_{23}=11$. $f_{23}=11$. $C_{23} = f_{23}$ **No action.**

For vertex j=5 , $f_{25}<0$ **(perform b)**

$\Delta_5 = \min(\Delta_2 - |f_{25}|) = \min(9,0) =0$

**L5** = {2$^-$ , 0}

**Scan 3. i**=3 **not labeled no action.**

**Scan 4. i**=4 Adjacent labels 1, 5. [ **j**=5] (j=1 is already labelled)

$C_{45}=7$. $f_{45}=0$.

$\Delta_{45} = C_{45} - f_{45} = 7 - 0 = 7$

$\Delta_5 = \min(\Delta_4 , \Delta_{45}) = \min(10, 7) =10$

**L5** = {4$^+$, 10}

**Scan 5. i**=5 Adjacent labels 2,3,6,4 [ **j**=3, **j**=6] (j=2 j=4 are already labelled)

$C_{56}=3$. $f_{56}=0$.

$\Delta_{56} = C_{56} - f_{56} = 3 - 0 = 3$

$\Delta_6 = \min(\Delta_5 , \Delta_{56}) = \min(10, 3) =3$

**L6** = {5$^+$, 3}

Since **vertex 6** **is t** $\Delta_t = 3$

----------------------------------

**Now all vertices are all labeled**

Find the path

**L6** = {5$^+$, 3} → **L5** = {4$^+$, 10} → **L4** = {1$^+$, 20} → **L1**

Thus one augmenting path is 1-4-5-6

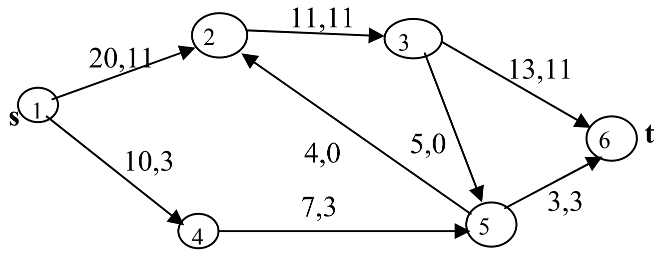
Add $\Delta_t=3$ to this path

$f_{14(new)} = f_{14(old)} + \Delta_t$

$f_{14} = 0 + 3$

$f_{45} = 0 + 3 = 3$

$f_{56} = 0 + 3 = 3$




## Remove all the labels. Start scanning

**2)** vertex 1 (s) is **labeled Ø**, 2,3,4,5,6 are **unlabeled**

**3) Scan 1. i**=1 ,Adjacent labels 2 and 4. [ **j**=2 and **j**=4]

$C_{12}=20$. $f_{12}=11$. (perform **a**)

For vertex j=2

$\Delta_{12} = C_{12} - f_{12} = 20 - 11=9$

$\Delta_2 = \Delta_{12} =9$.

**L2** = {1$^+$, 9}

For vertex j=4

$C_{14}=10$. $f_{14}=3$.

$\Delta_{14} = C_{14} - f_{14} = 10 - 3=7$

$\Delta_4 = \Delta_{14} =7$.

**L4** = {1$^+$, 7}

**Scan 2. i**=2 , Adjacent labels 1, 3 and 5. [ **j**=3 and **j**=5] (j=1 is already labelled)

For vertex j=3,

$C_{23}=11$. $f_{23}=11$. $C_{23} = f_{23}$ **No action.**

For vertex j=5 , $f_{25}<0$ **(perform b)**

$\Delta_5 = \min(\Delta_2 - |f_{25}|) = \min(9,0) =0$

**L5** = {2$^-$ , 0}

**Scan 3. i**=3 **not labeled no action.**

**Scan 4. i**=4 Adjacent labels 1, 5. [ **j**=5] (j=1 is already labelled)

$C_{45}=7$. $f_{45}=0$.

$\Delta_{45} = C_{45} - f_{45} = 7 - 0 = 7$

$\Delta_5 = \min(\Delta_4, \Delta_{45}) = \min(10, 7) = 10$

**L5** = {$4^+$, 10}

**Scan 5.** **i**=5 Adjacent labels 2,3,6,4 [ **j**=3, j=6]
(j=2 j=4 are already labelled)

$C_{56}=3$. $f_{56}=0$.

$\Delta_{56} = C_{56} - f_{56} = 3 - 0 = 3$

$\Delta_6 = \min(\Delta_5, \Delta_{56}) = \min(10, 3) = 3$

**L6** = {$5^+$, 3}

**Solution 1)** $C_{12}=20$, $C_{23}=11$, $C_{36}=13$, $C_{35}=5$, $C_{14}=10$, C

**Scan 4.** **i**=4 ,Adjacent labels 1, 5 and 6. [ **j**=6 ]
(j=2 and j=5 are already scanned)

$C_{36}=13$. $f_{36}=0$.

$\Delta_{36} = C_{36} - f_{36} = 13 - 0 = 13$

$\Delta_6 = \min(\Delta_3, \Delta_{36}) = \min(11, 13) = 11$

**L6** = {$3^+$, 11}

Since **L6** is **t**

For vertex j=5 , $f_{25} < 0$ **(perform b)**

$\Delta_5 = \min(\Delta_2 - | f_{25} | ) = \min(20,0) = 0$

**L5** = {$2^-$ , 0}

$\Delta_4 = \Delta_{14} = 10$.

**L4** = {$1^+$, 10}

vv

Scan 2 and 4.

Second Number: given flow ($f_{i,j}$)
S: source t: target
Path: sequence of edges in a diagraph

**Flow augmenting path:** Paths from S to t.
Examples: Path 1=(1-2-3-6) Path 2=(1-4-5-6)
Path 3=(1-4-5-3-6)

**Forward edge:** If the direction of path is the same as the direction of edge it is called forward edge.

**Backward edge:** If the direction of path is the opposite of the direction of edge it is called forward edge.

Path 1: 1-2, 2-3, 3-6 all forward edges

Path 3: 1-4, 4-5, 3-6 forward edges **5,3** backward edge

$C_{ij}$=the capacity of edge from i to j
$f_{ij}$=The value of current flow from i to j.
$\Delta_{ij}$=possible additional flow from edge i to j.
$\Delta_{ij} = C_{ij} - f_{ij}$

$\Delta_{12}$ =20-5=15, $\Delta_{23}$ =11-8=3, $\Delta_{34}$ =13-6=7,
$\Delta_{14}$ =10-4=6, $\Delta_{45}$ =7-4=3, $\Delta_{56}$=3-3=0,
$\Delta_{35}$=5-2 =3.

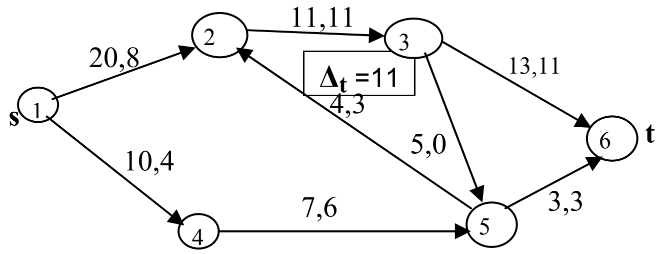
**Maximum Flow**: Maximum possible flow from **s** to **t**

**Kirchof's rule: Incoming flow=Outgoing flow**

Example: for vertex 2, 5,3 incoming flow.
8:outgoing flow.. 5+3=8

Possible additional flow in path 1
We can increase maximum flow by 3 because the edge 2,3 allows only 3.

No additional flow is possible in path 2, because $\Delta_{56}$=0, additional flow is possible in path 3.




**Ford-Fulkerson Algorithm for Maximum Flow**

**ALGORـTHM FORD-FULKERSON**

[G = (V, *E),* vertices l (= *s),* • • • , *n* (= *t),* edges (;', *j),* Cy]

This algorithm computes the maximum flow in a network G with source *s,* sink (, aý capacities Cy > O of the edges (;', *j).*

INPUT: *n, s* = l, *t* = n, edges (;', *j)* of G, Cy

OUTPUT: Maximum flow *f* in G

1. Assign an initial flow *f y* (for instance, *f y* = O for ali edges), compute *f.*
2. Label *s* by 0. Mark the other vertices *"unlabeled."*

***J.* l'lllU a laü^l^U v^/ha/a. t tilUL lýuo llüt J^l Uüülý a^^mý^u. 0^0.11**

For every unlabeled adjacent vertex *j,* if Cy > *f\p* compute

Ay = Cy - *f*y **and** A, A if (• = 1 ' ü im in (A,, Ay) if ;• > 1 and label7 with a *"fonvard label"* (i"1', A,); or if *f*^ > O, compute A, == min(A,, *f,,)* and label *j* by a "backward label" (;~, Aj). If no such *j* exists then OUTPUT *f*. Stop [^" ;5 r/îe *maximum flow.}* Else continue (that is, go to Step 4).

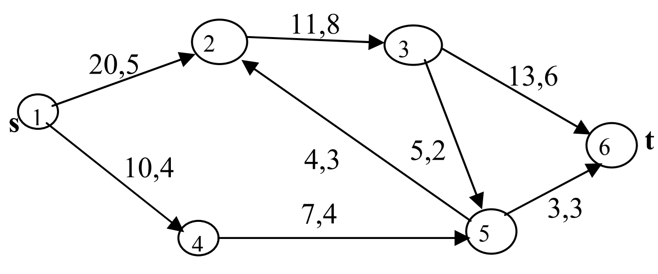
4. Repeat Step 3 until *t* is reached.

*[This gives a flow augmenting path P: s —>• t.}*
If it is impossible to reach *t* then OUTPUT *f*. Stop *[f is the maximum flow.}* Else continue (that is, go to Step 5).

5. Backtrack the path *P*, using the labels.
6. Using *P*, augment the existing flow by a(. Set *f* = *f* + a(.
7. Remove ali labels from vertices 2, • • • , *n. Go* to Step 3.

End FORD-FULKERSON

**A Network** is a diagraph in which each edge has assigned to it a capacity (maximum flow)




Graphs and Combinatorial Optimization

## 21.7 Ford-Fulkerson Algorithm for Maximum Flow

Flow augmenting paths, as discussed in the last section, are used as the basic tool ini Ford-Fulkerson algorithm in Table 21.8 in which a given flow (for instance, zero flofl ali edges) is increased until it is maximum. The algorithm accomplishes the increase a stepwise construction of flow augmenting paths, one at a time, until no further **sý** paths can be constructed, which happens precisely when the flow is maximum.

**Table 21.8**

**Ford-Fulkerson Algorithm for Maximum Flow**

**ALGORـTHM FORD-FULKERSON**

[G = (V, *E),* vertices l (= *s),* • • • , *n* (= *t),* edges (;', *j),* Cy]

This algorithm computes the maximum flow in a network G with source *s,* sink (, aý capacities Cy > O of the edges (;', *j).*

INPUT: *n, s* = l, *t* = n, edges (;', *j)* of G, Cy

OUTPUT: Maximum flow *f* in G

1. Assign an initial flow *f y* (for instance, *f y* = O for ali edges), compute *f.*
2. Label *s* by 0. Mark the other vertices *"unlabeled."*

***J.* l'lllU a laü^l^U v^/ha/a. t tilUL lýuo llüt J^l Uüülý a^^mý^u. 0^0.11**

For every unlabeled adjacent vertex *j,* if Cy > *f\p* compute

A if (• = l

'ü

Ay = Cy - *f*y **and** A,

imin (A,, Ay) if ;• > l

and label7 with a *"fonvard label"* (i"[1]', A,); or if *f*^ > O, compute

A, == min(A,, *f,,)*

and label *j* by a "backward label" (;~, Aj).

If no such *j* exists then OUTPUT *f.* Stop

[^" ;5 r/îe *maximum flow.}*

Else continue (that is, go to Step 4).

4. Repeat Step 3 until *t* is reached.

*[This gives a flow augmenting path P: s —>• t.}*

If it is impossible to reach *t* then OUTPUT *f.* Stop

*[f is the maximum flow.}*

Else continue (that is, go to Step 5).

5. Backtrack the path *P,* using the labels.
6. Using *P,* augment the existing flow by a(. Set *f* = *f* + a(.
7. Remove ali labels from vertices 2, • • • , *n. Go* to Step 3.

End FORD-FULKERSON



**Table21.9**