Push-relabel

Maximum Flow Algorithm

General concepts

- The **push–relabel algorithm** is an algorithm for computing maximum flows.
- The push–relabel algorithm is considered one of the most efficient maximum flow algorithms. The generic algorithm has a strongly polynomial $O(V^2E)$ time complexity.

Definitions

- Consider a flow network $G(V, E)$ with a pair of distinct vertices s and t designated as the source and the sink, respectively. For each edge $(u, v) \in E$, $c(u, v) \geq 0$denotes its capacity; if $(u, v) \notin E$, we assume that $c(u, v) = 0$. A flow on G is a function f: $V \times V \rightarrow \mathbf{R}$ satisfying the following conditions:
  - Capacity constraints

    $f{:}(u, v) \leq c(u, v) \quad \forall u, v \in V$
  - Skew symmetry:

    $f(u, v) = -f(v, u) \quad \forall u, v \in V$
  - Flow conservation

    $\sum_{v \in V} f(v, u) = 0 \quad \forall u \in V \setminus \{s, t\}$
- The push–relabel algorithm introduces the concept of *preflows*. A preflow is a function with a definition almost identical to that of a flow except that it relaxes the flow conservation condition. Instead of requiring strict flow balance at vertices other than s and t, it allows them to carry positive excesses.

Operations

- **Push**

➢ The push operation applies on an admissible out-edge (u, v) of an active vertex u in G_f. It moves $\min\{e(u), c_f(u, v)\}$ units of flow from u to v.

push(u, v):

> ***assert e[u] > 0 and h[u] == h[v] + 1***
>
> ***Δ = min(e[u], c[u][v] - f[u][v])***
>
> ***f[u][v] += Δ***
>
> ***f[v][u] -= Δ***
>
> ***e[u] -= Δ***
>
> ***e[v] += Δ***

Operations

• Relabel

➢ The relabel operation applies on an active vertex u without any admissible out-edges in G_f. It modifies $h(u)$ to the minimum value such that an admissible out-edge is created. Note that this always increases $h(u)$ and never creates a steep edge (an edge (u, v) such that $c_f(u, v) > 0$, and $h(u) > h(v) + 1$).

relabel(u):

assert e[u] > 0 and h[u] <= h[v] ∀v such that f[u][v] < c[u][v]

h[u] = min(h[v] ∀v such that f[u][v] < c[u][v]) + 1

Obs. After a push or relabel operation, h remains a valid height function with respect to f.

Push–relabel algorithm

- At initialization, the algorithm fulfills this requirement by creating a preflow f that saturates all out-edges of s, after which $h(u) = 0$ is trivially valid for all $v \in V \setminus \{s, t\}$.
- After initialization, the algorithm repeatedly executes an applicable push or relabel operation until no such operations apply, at which point the preflow has been converted into a maximum flow.

push-relabel(G(V, E), s, t):

create a preflow f that saturates all out-edges of s

let h[u] = 0 ∀v ∈V

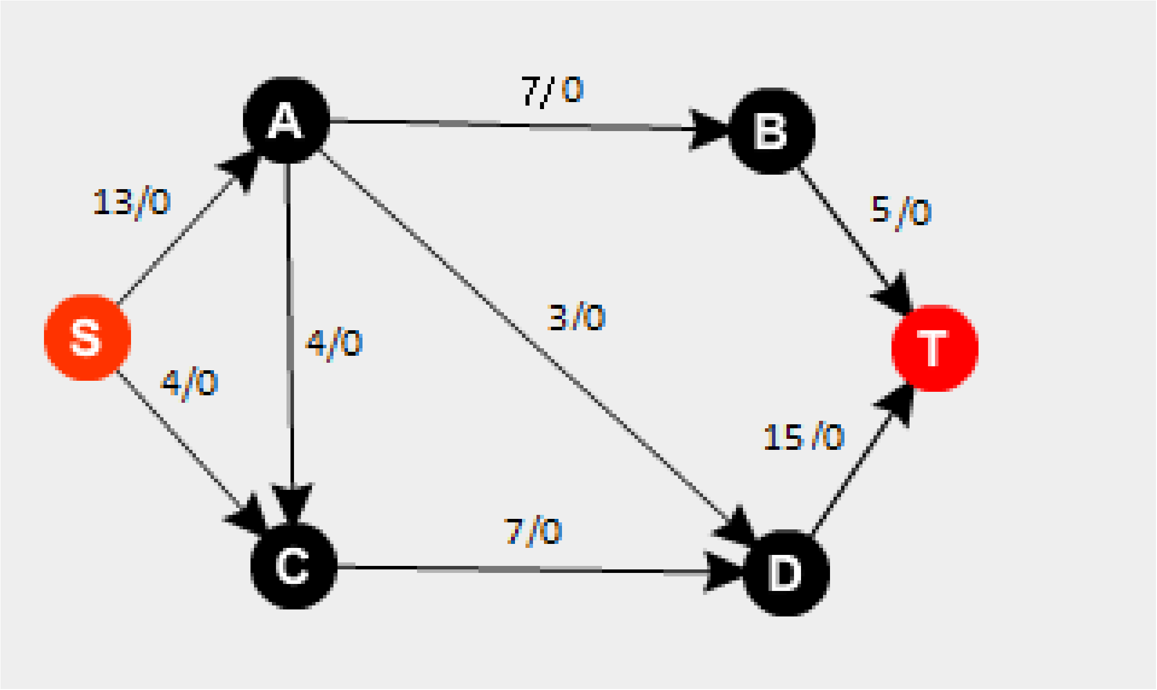
while there is an applicable push or relabel operation

execute the operation

Example

Initialization

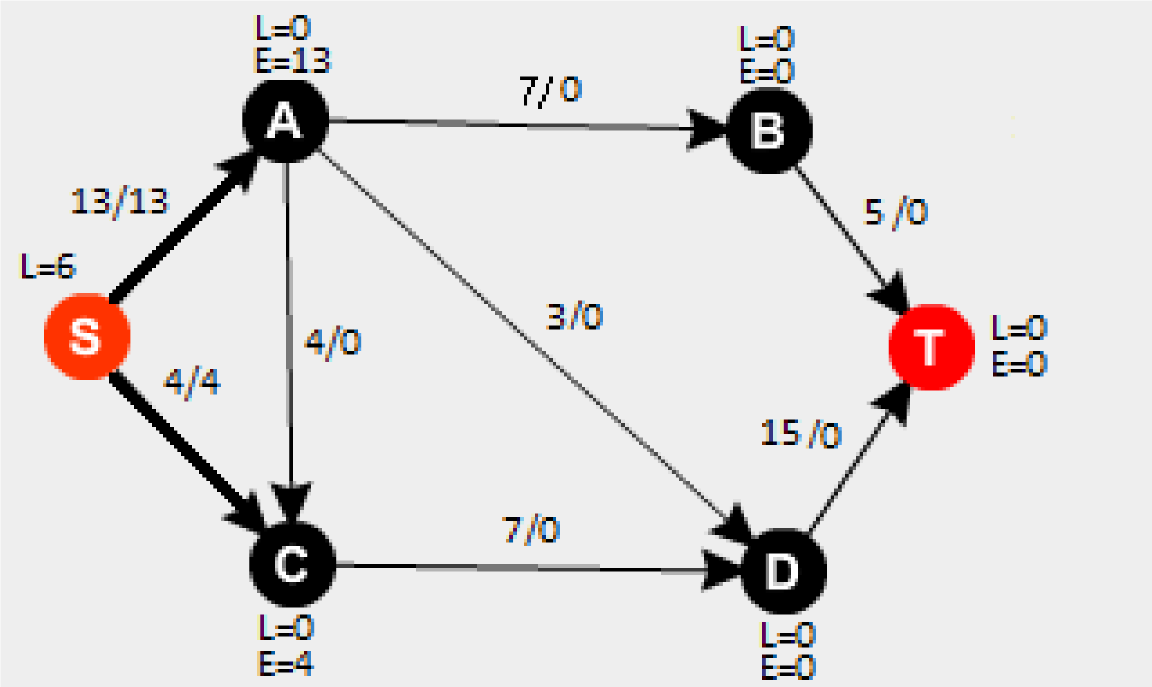
The prefolow function pushes as much flow as possible through the edges from the source s.The edges are (s,a) and (s,c) and the preflow is 13,respectively 4.All other flows are set to 0



Example

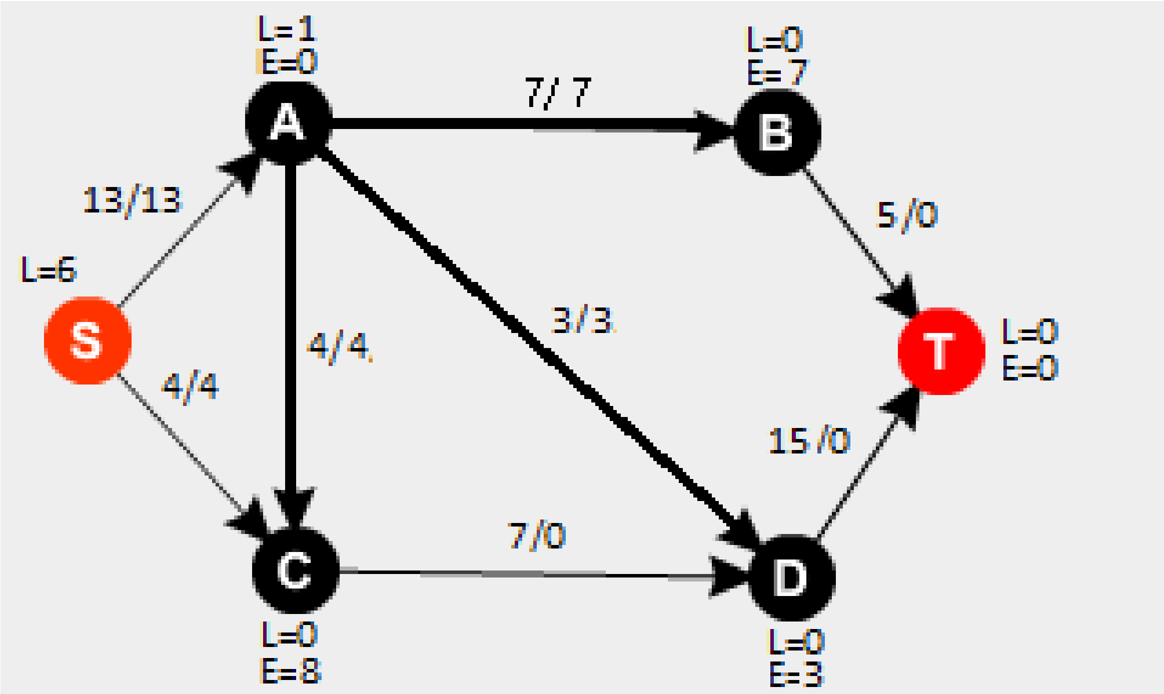
Step 1:

The level of s is set to 6 (number of nodes). For all other nodes the level is set to 0.
The excess of s is set to infinit. The excess of a is set to 13 oa the difference between incoming flow and outcoming flow.



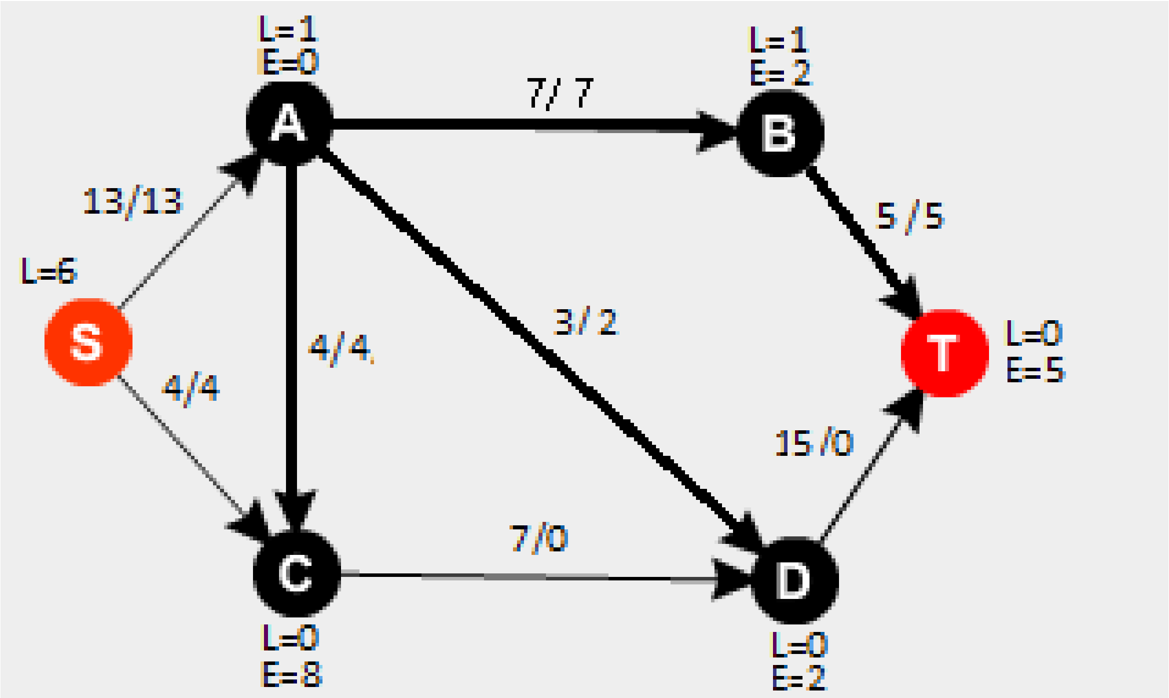
Example

Step 2:
Relabel a: L(a) = 1
Push(a,b):F(a,b) = 7;E(a) = 6;E(b) = 7
Push(a,c) : F(a,c) = 4;E(a) = 2;E(c) = 8
Push(a,d):F(a,d) = 2; E(a) = 0;E(d) = 2



Example

Step 3:
Relabel a: L(b) = 1
Push(b,t):F(b,t) = 7;E(b) = 2;E(t) = 5

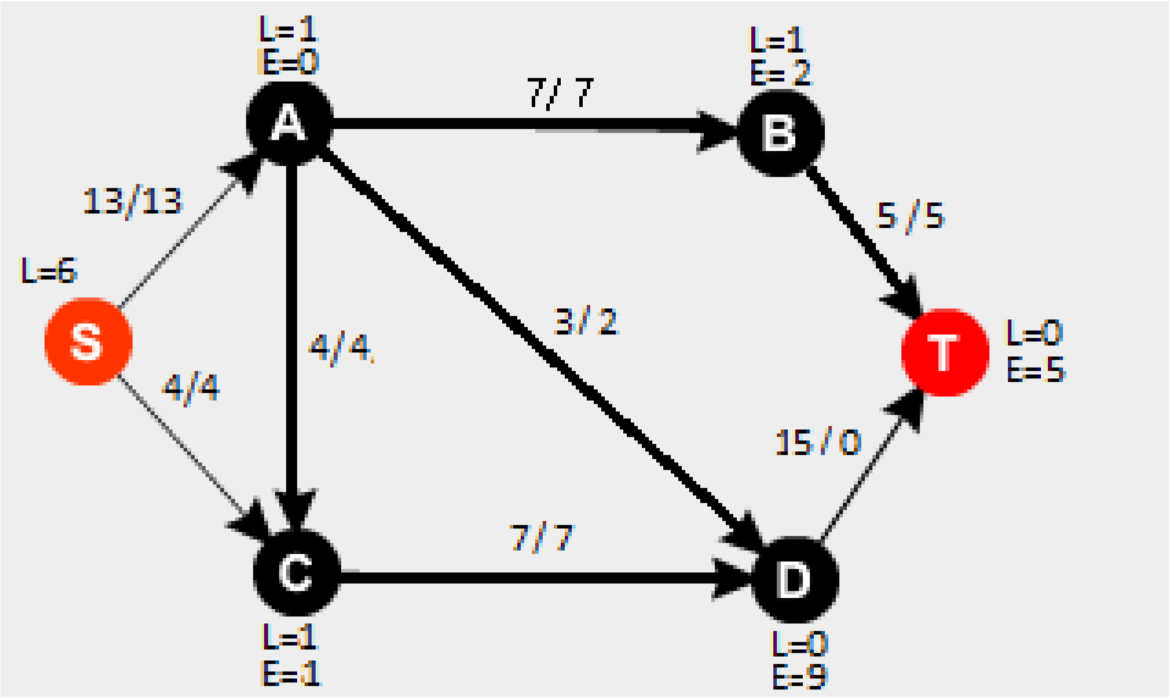


Example

Step 4:

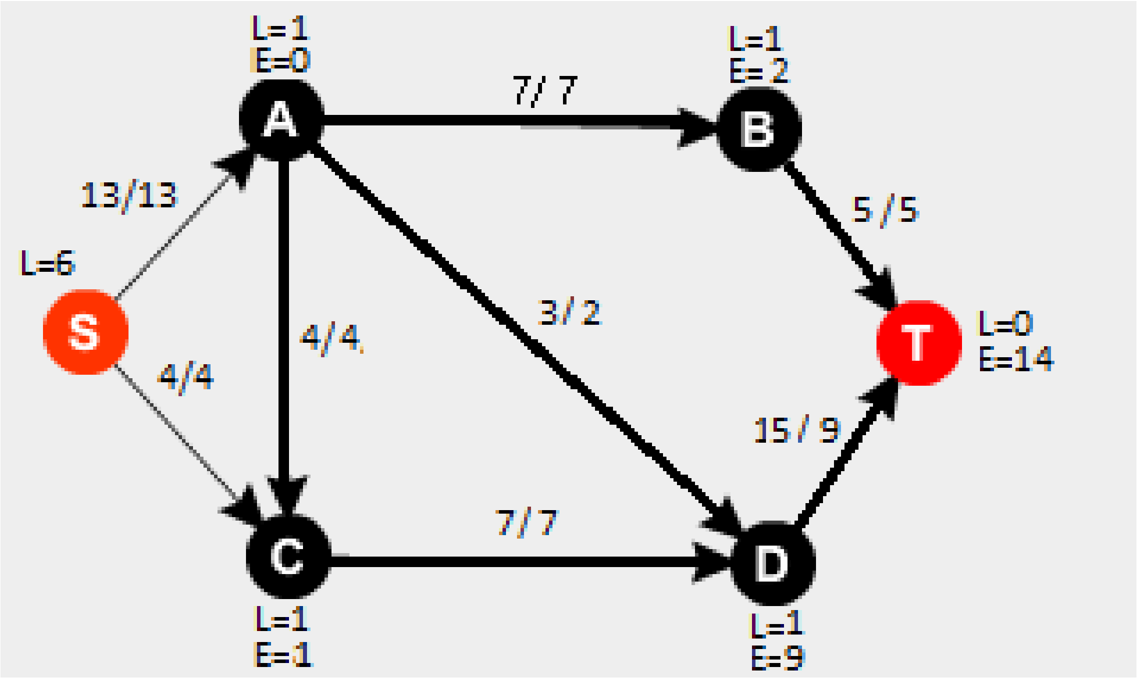
Relabel c: L(c) = 1

Push(c,d):F(c,d) = 7;E(c) = 1;E(d) = 9



Example

Step 5:
Relabel d: L(d) = 1
Push(d,t):F(d,t) = 7;E(d) = 4 ;E(t) = 14

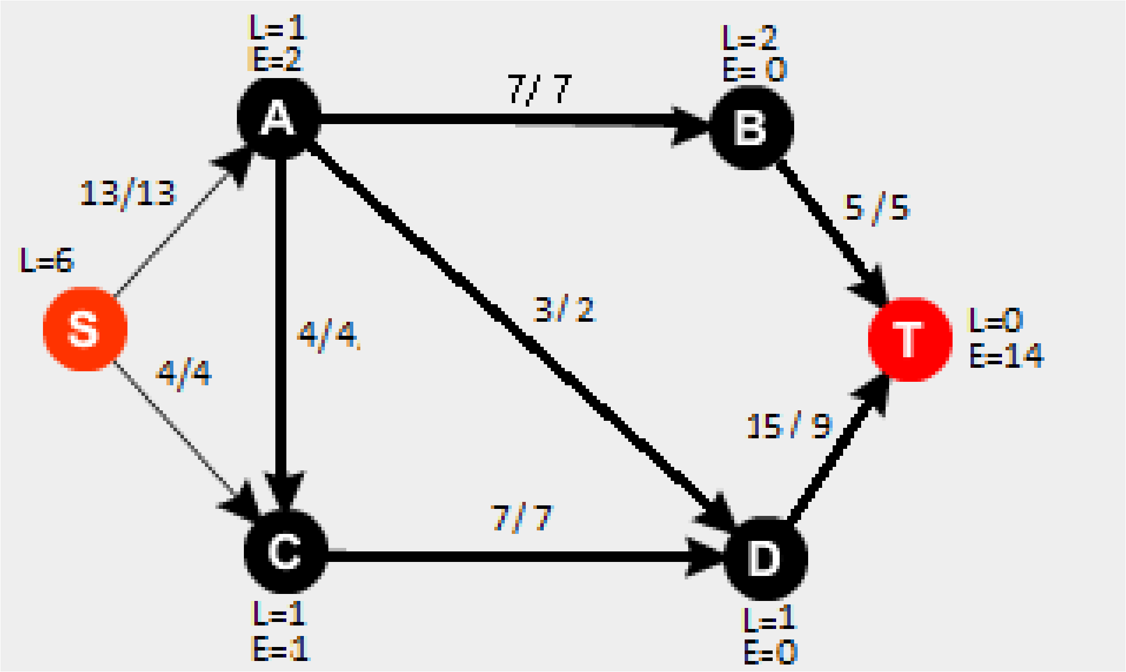


Example

Step 6:

Relabel b: L(b) = 2

E(b) = 0 ,E(a) = 2 – The excess is transferred downwards to a even if the edge is upwards.The excess that is transferred came at a previuus step from the source but can not contribute to the flow because the edge (a,b) is saturated, so it is transferred to node a



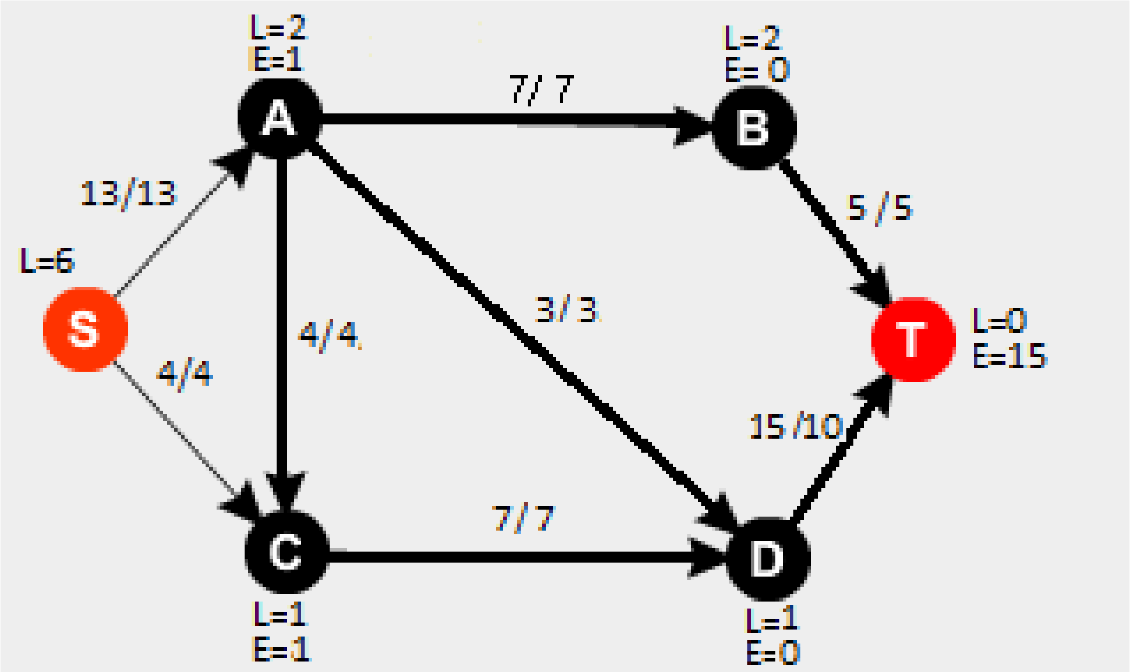
Example

Step 7:

Relabel a: L(a) = 2-Thes is done because a hs excess

Push(a,d) : F(a,d) = 3;E(a) = 1;E(c) = 1

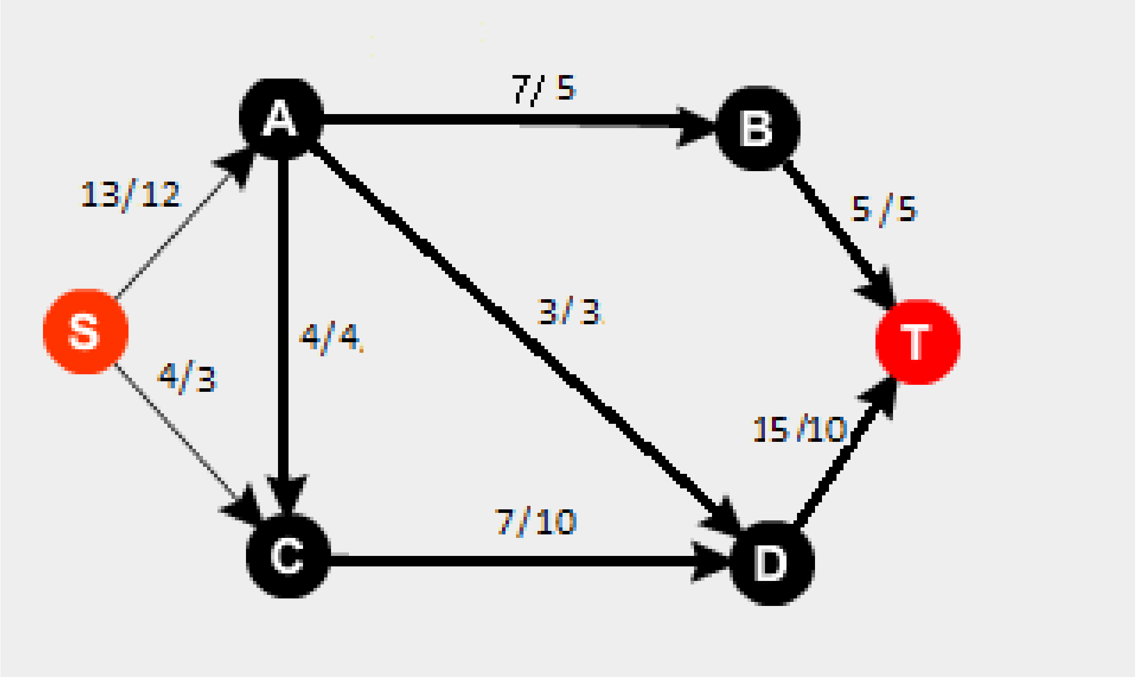
Push(d,t):F(d,t) = 2; E(d) = 10;E(t) = 15



Example

Final result:

Maximum Flow :15



Implementation

Argorithm implemention can be found here