Maximum-Flow Problem

YAO ZHAO

OUTLINE

- Ford-Fulkerson Algorithm
- Edmonds-Karp Algorithm
- Dinic Algorithm
- A Question

Ford-Fulkerson Review

```
Augment(f, c, P) {
    b ← bottleneck(P)
    foreach e ∈ P {
        if  (e ∈ E)  f(e)  ← f(e)  + b
        else            f(eᴿ)← f(eᴿ)  - b
    }
    return f
}
```

```
Ford-Fulkerson(G, s, t, c) {
    foreach e ∈ E  f(e) ← 0
    Gf ← residual graph

    while (there exists augmenting path P) {
        f ← Augment(f, c, P)
        update Gf
    }
    return f
}
```

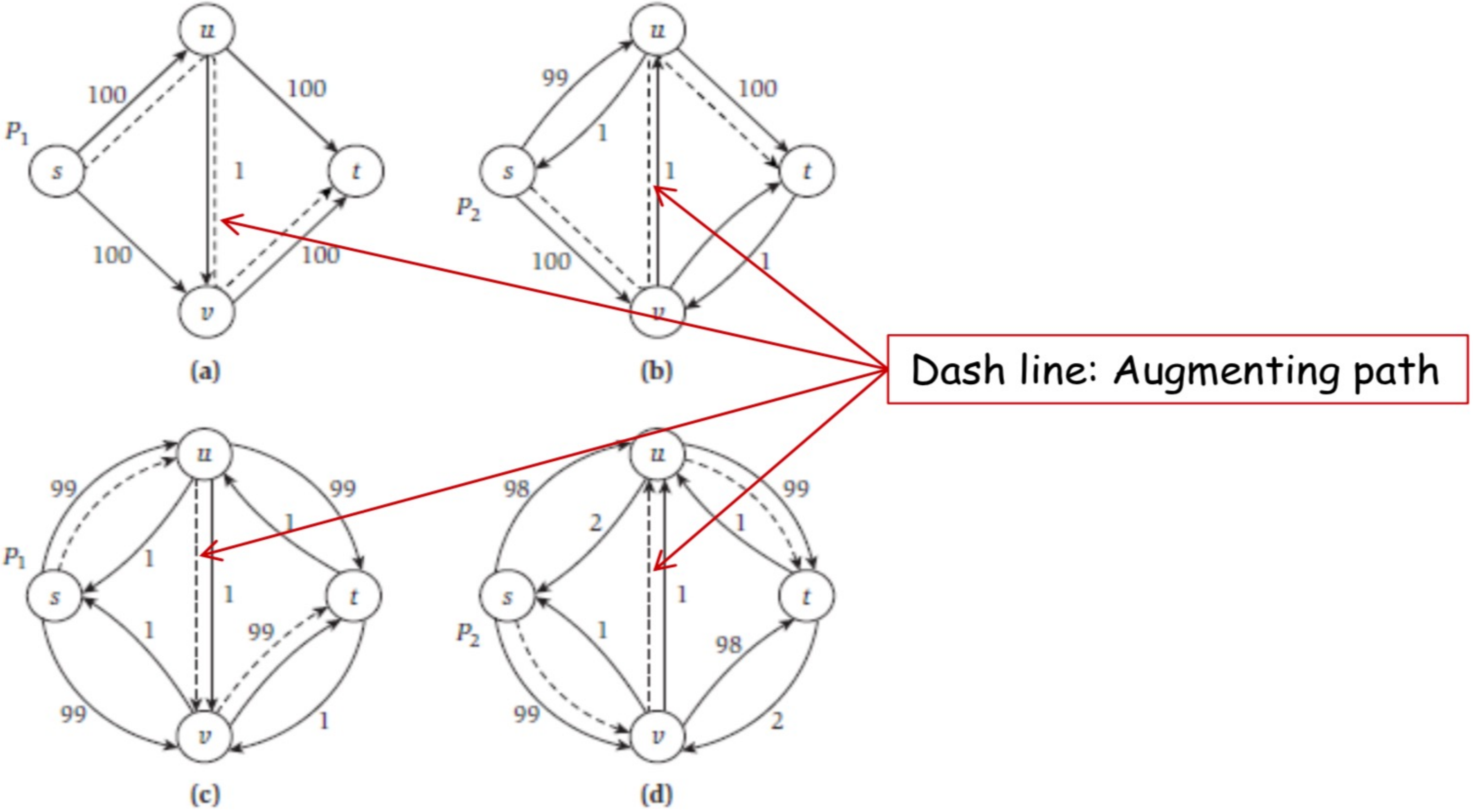


Figure Parts (a) through (d) depict four iterations of the Ford-Fulkerson Algorithm using a bad choice of augmenting paths: The augmentations alternate between the path P_1 through the nodes s, u, v, t in order and the path P_2 through the nodes s, v, u, t in order.

Edmonds-Karp (Shortest augmenting path)

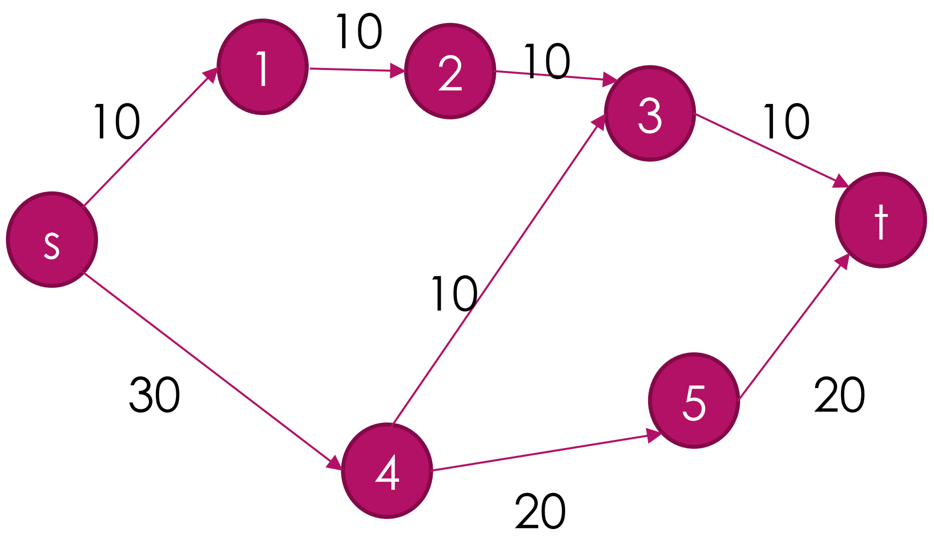
- Always find the shortest augmenting path
- Complexity improved to $O(|V||E|^2)$

BFS(G_f) and can find a path P from s to t

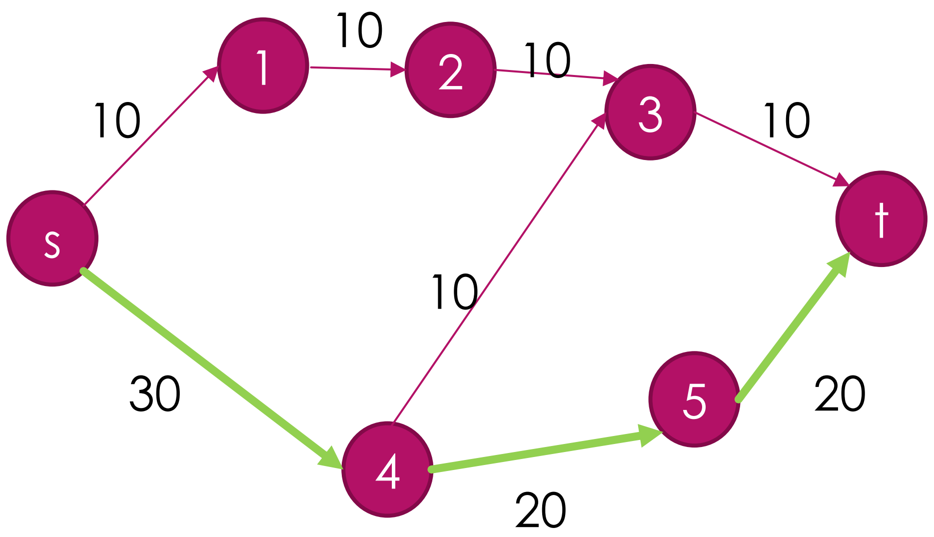
```
Ford-Fulkerson(G, s, t, c) {
   foreach e ∈ E  f(e) ← 0
   Gf ← residual graph

   while (there exists augmenting path P) {
      f ← Augment(f, c, P)
      update Gf
   }
   return f
}
```

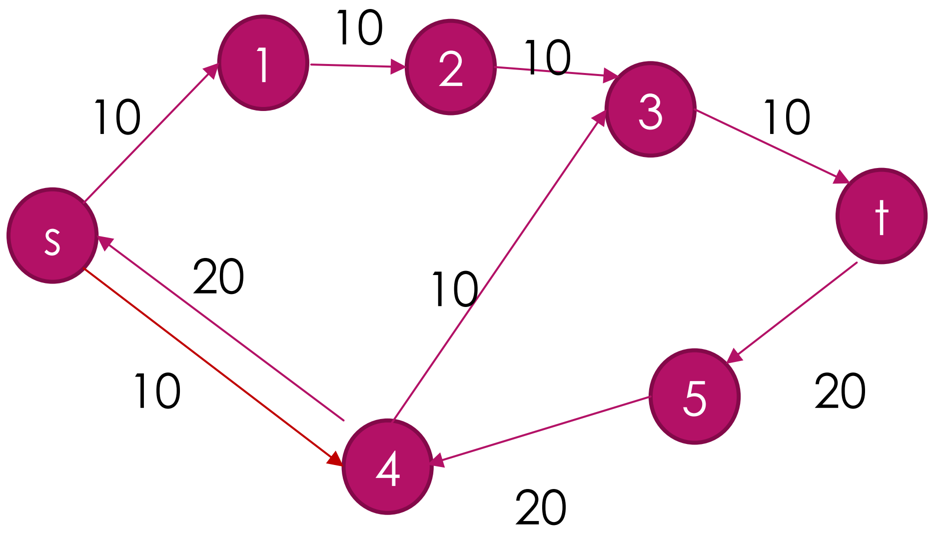
Edmonds-Karp process(1)



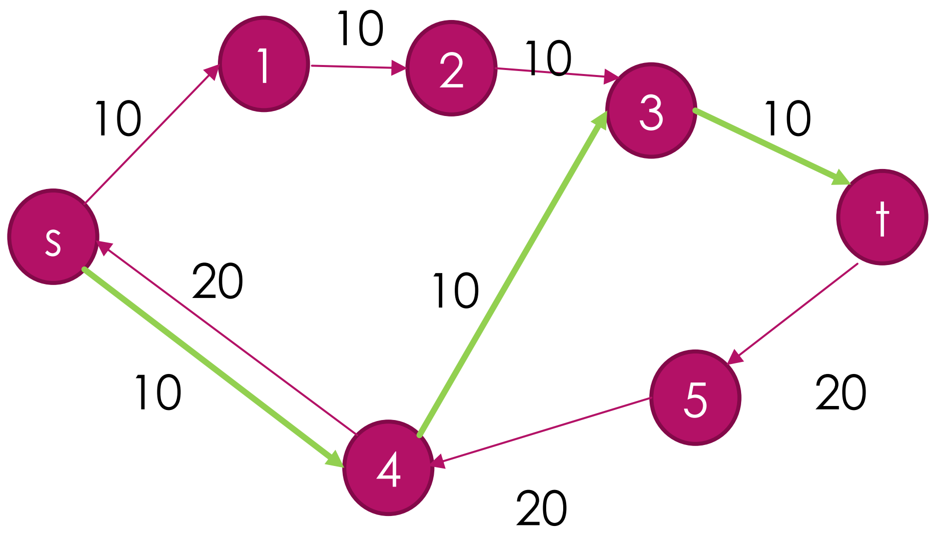
BFS first time



Edmonds-Karp process(2)

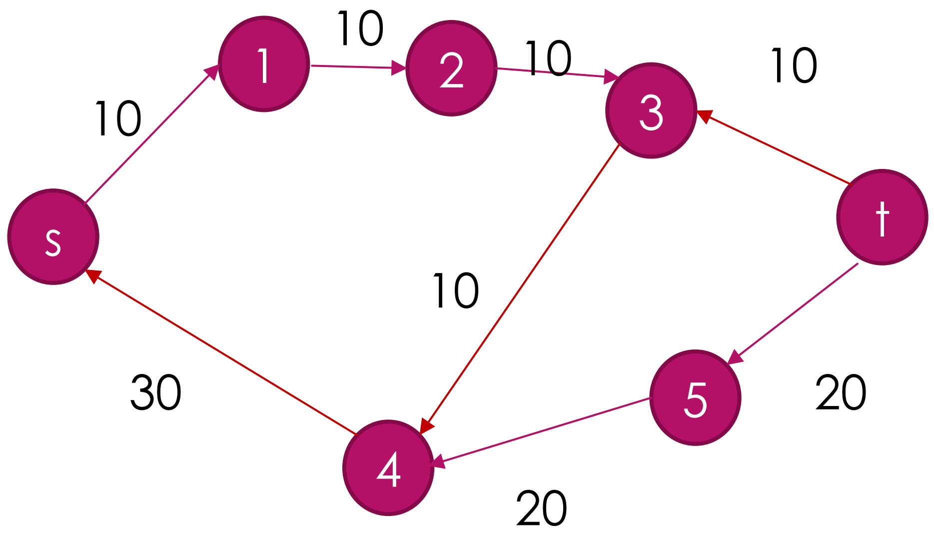


BFS second time



Edmonds-Karp process(3)

Stop
Max flow is 30



Every time BFS starts from s to t -- a waste of time?

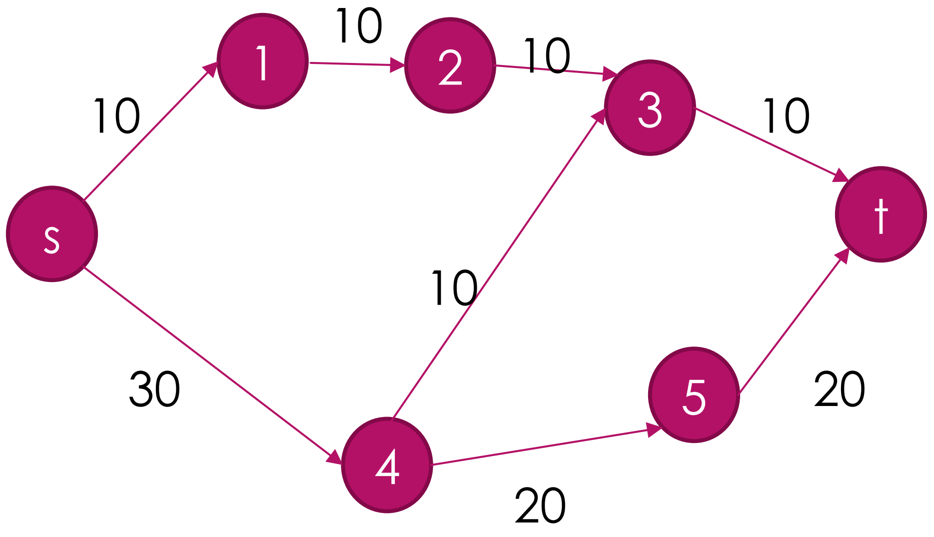
Dinic

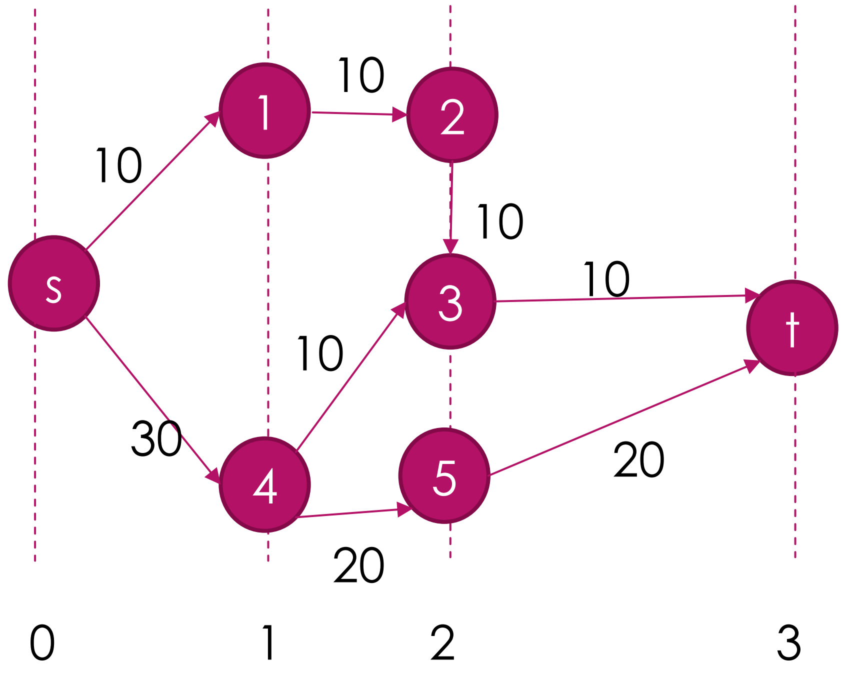
- Further optimized Edmonds-Karp algorithm
- DFS is used to find augmented paths (In a hierarchical network, DFS always find a shortest path)

Dinic process(1)

Initial residual graph

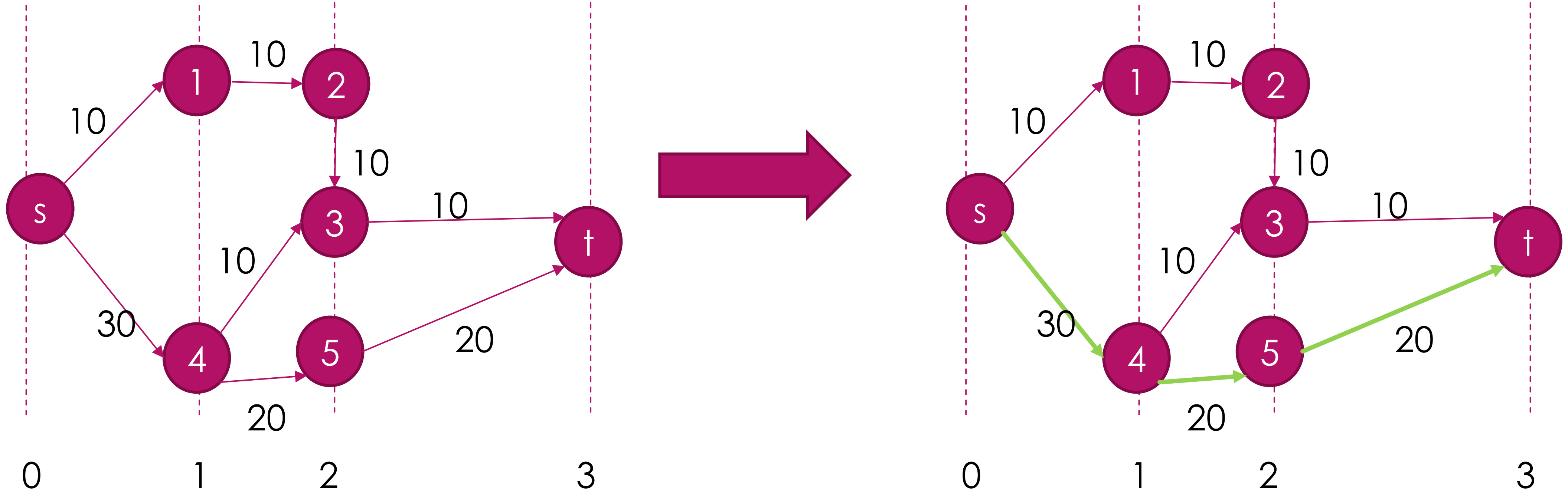
1、construct residual graph and hierarchical network， If the sink is not in the hierarchical network, stop; Otherwise, proceed to step 2.





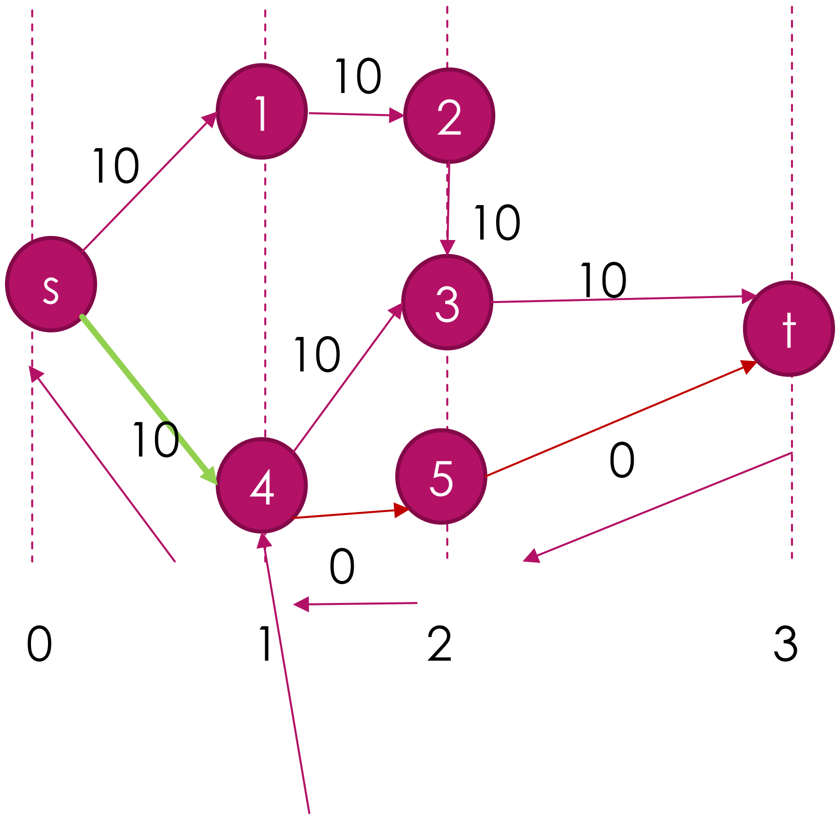
Dinic process(2)

2、 Using DFS algorithm, proceed from the vertex of layer i to the vertex of layer i +1, if reach the sink point t， means find an augmented path. If cannot reach t, Go back to step 1 （P.11）

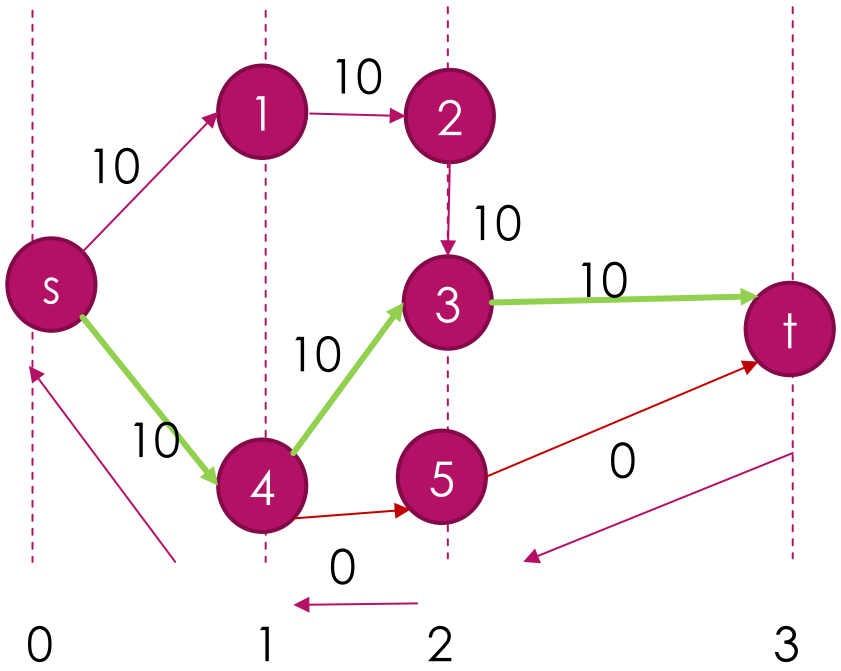


Dinic process(3)

3、Go back and update the capacity value. When return to layer l, and you find that you have another edge can reach layer l + 1. If the path reach t, you find another augmented path.



DFS return here，update the capacity of edge.

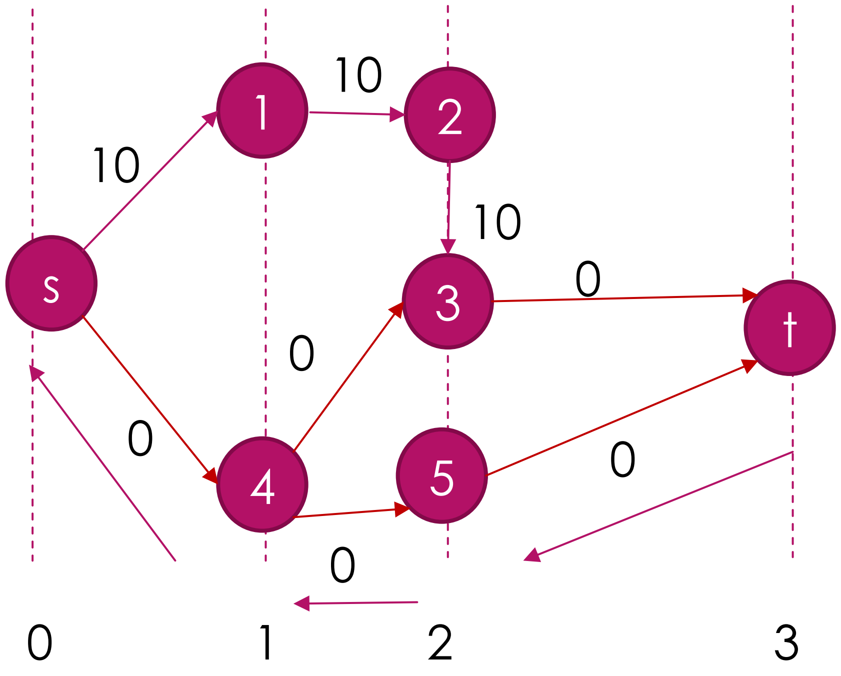


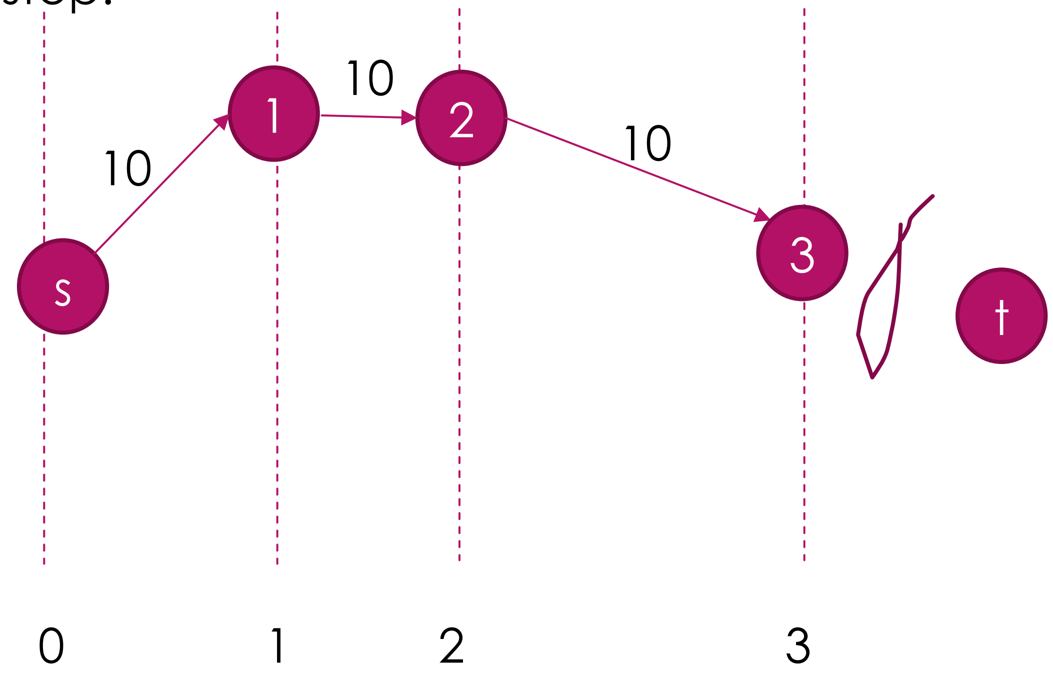
Find another path from the layer 1.

Dinic process(4)

4、 Step back, update the capacity value, and if no new augmented path can be found until step back to point s, the DFS process ends. Go back to step 2 (P.12).

In step 2: can't reach t, go back to step 1 (P.11), then stop.





Construct a new hierarchical network

The sink point doesn't join to the new hierarchical network, stop.