네트워크 플로우 (참고)

최백준 choi@startlink.io

Dinic’s Algorithm

Dinic’s Algorithm

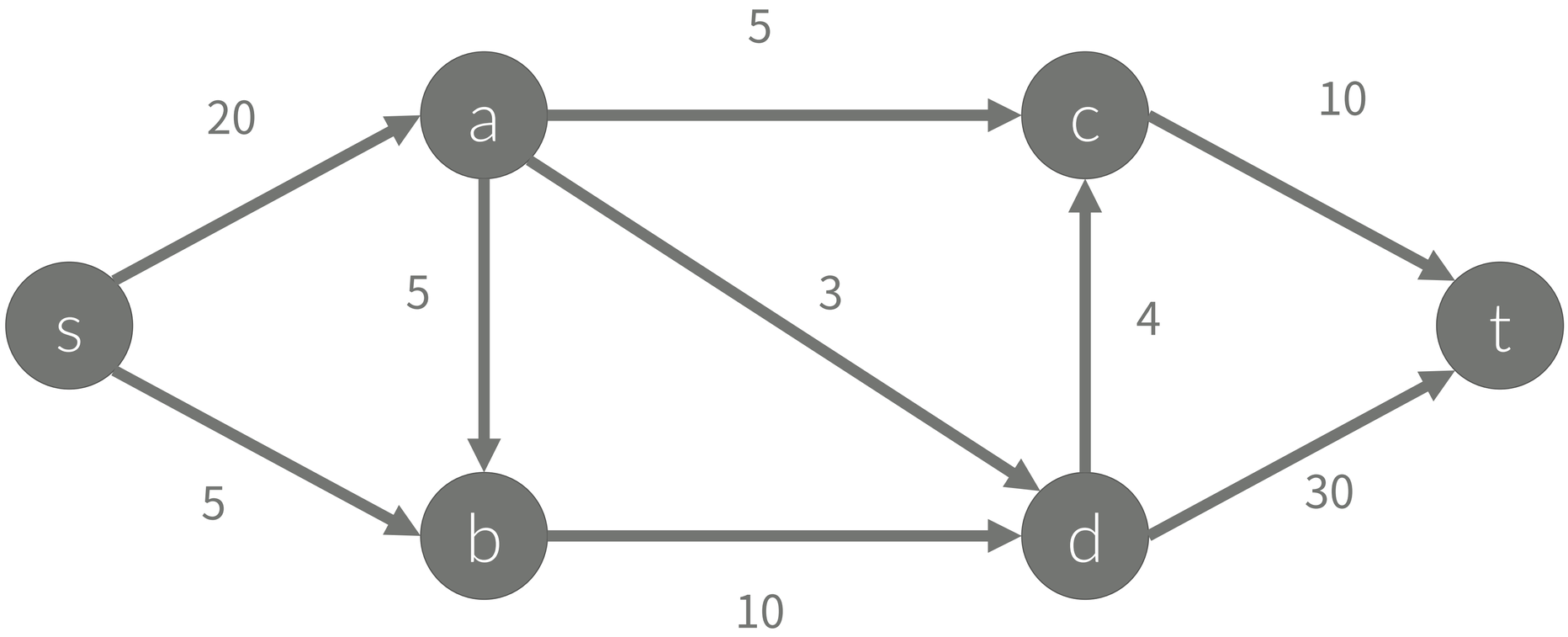
Maximum Flow

- BFS를 이용해서 flow가 더 있는지 알아ㅂㄴ다.
- 이를 위해 BFS로 Level Graph를 만든다.
- Level Graph는 소스에서 각 정점까지 가는 간선의 개수와 같은 의미를 가지고 잇다.
- Level Graph를 이용해 여러 개의 flow를 한 번에 흘려준다.
- 시간 복잡도는 $O(V^2E)$

Dinic’s Algorithm

Maximum Flow

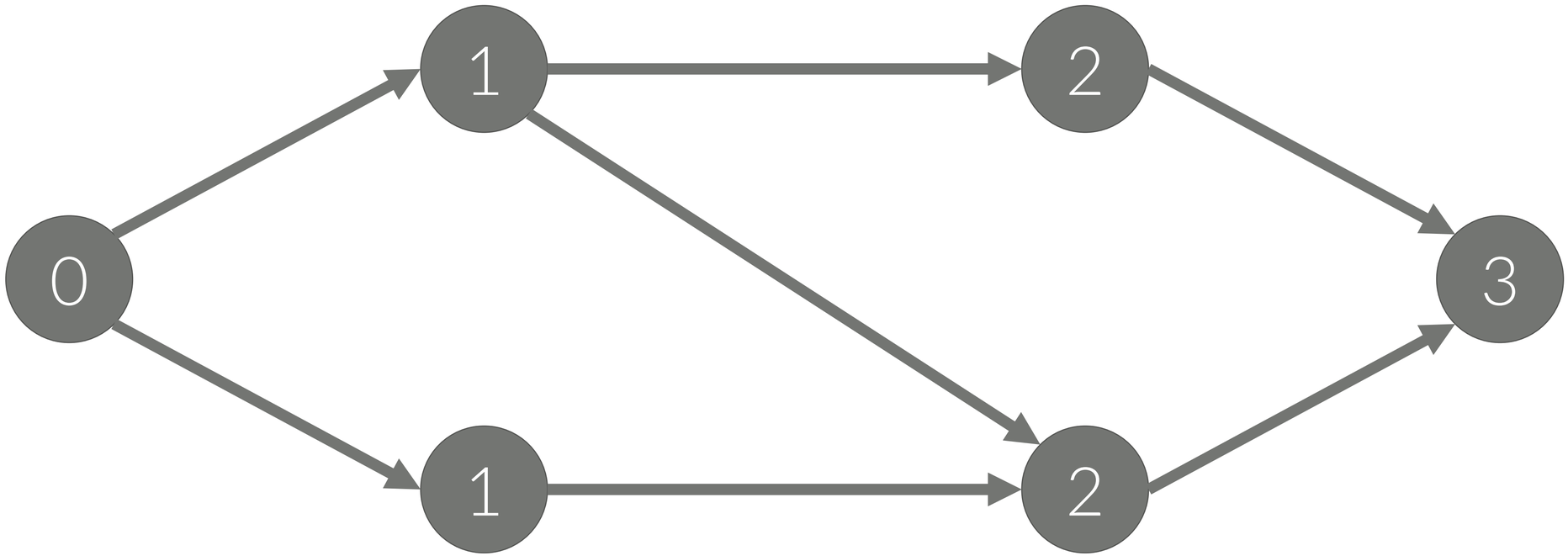
- Network Graph



Dinic’s Algorithm

Maximum Flow

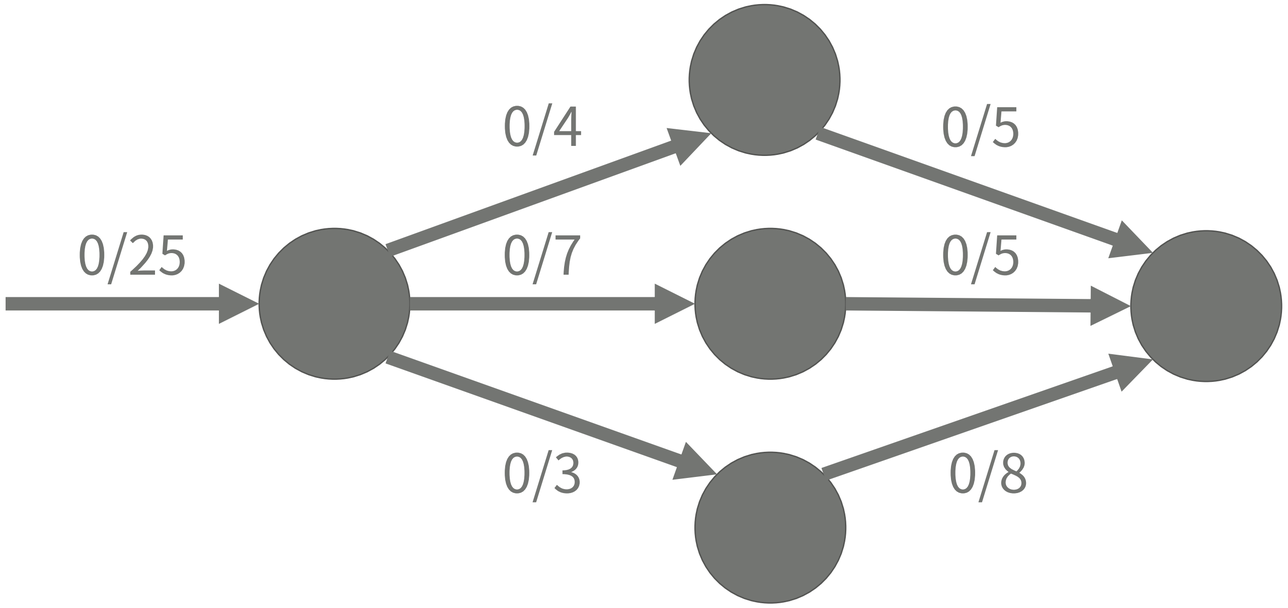
- Level Graph



Dinic’s Algorithm

Maximum Flow

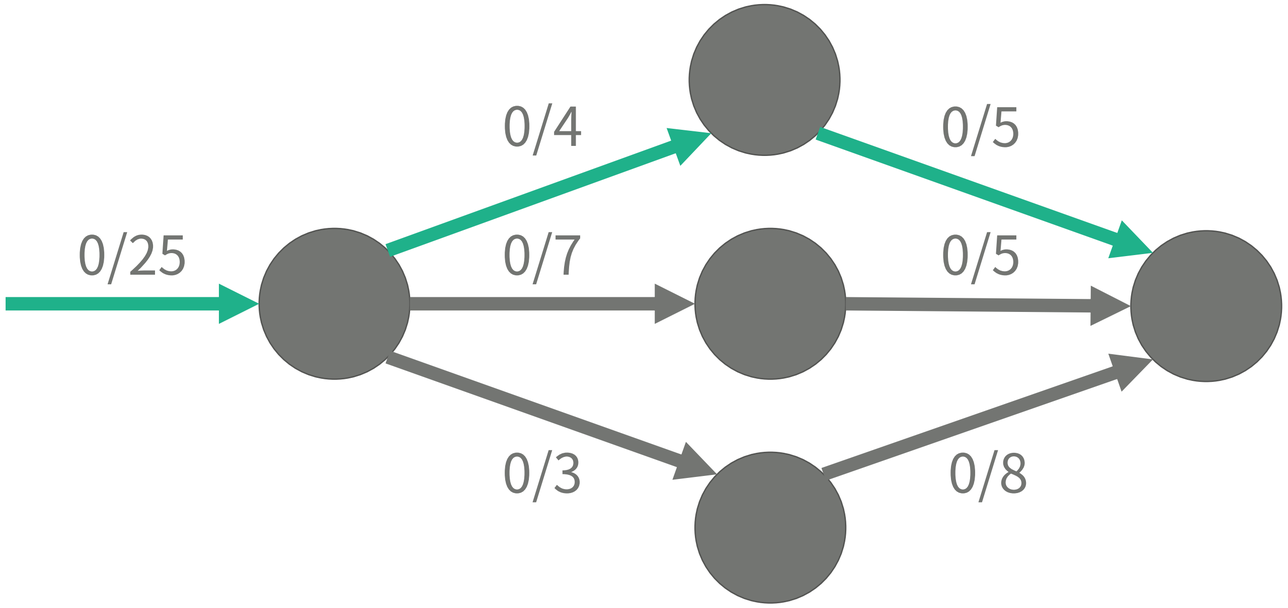
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

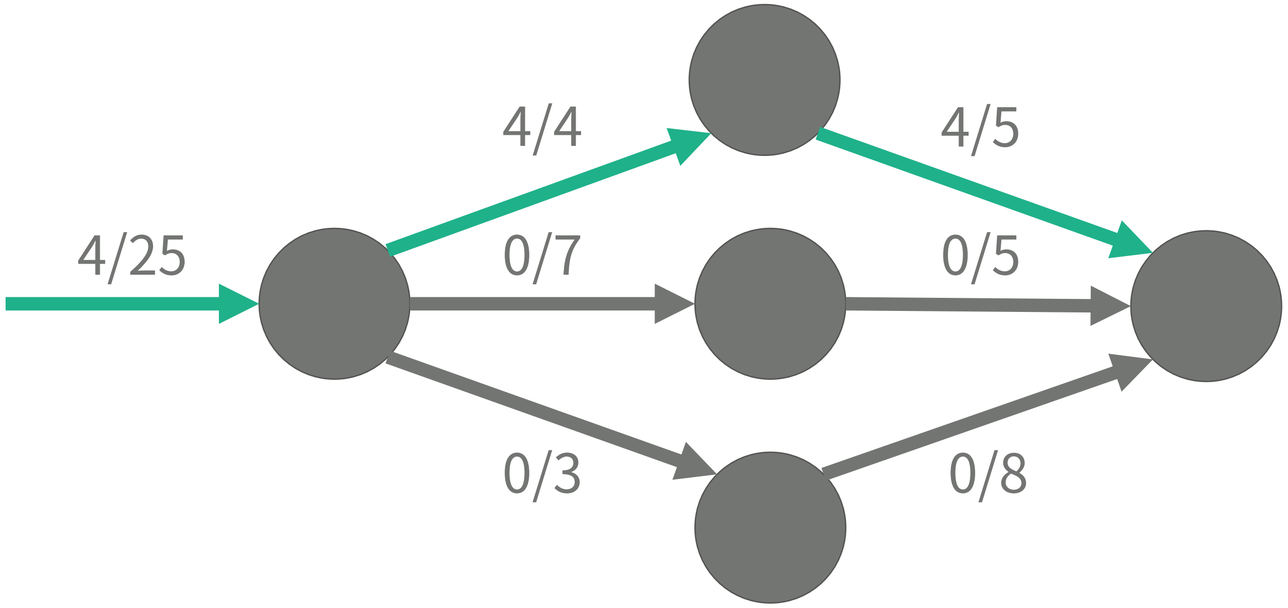
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

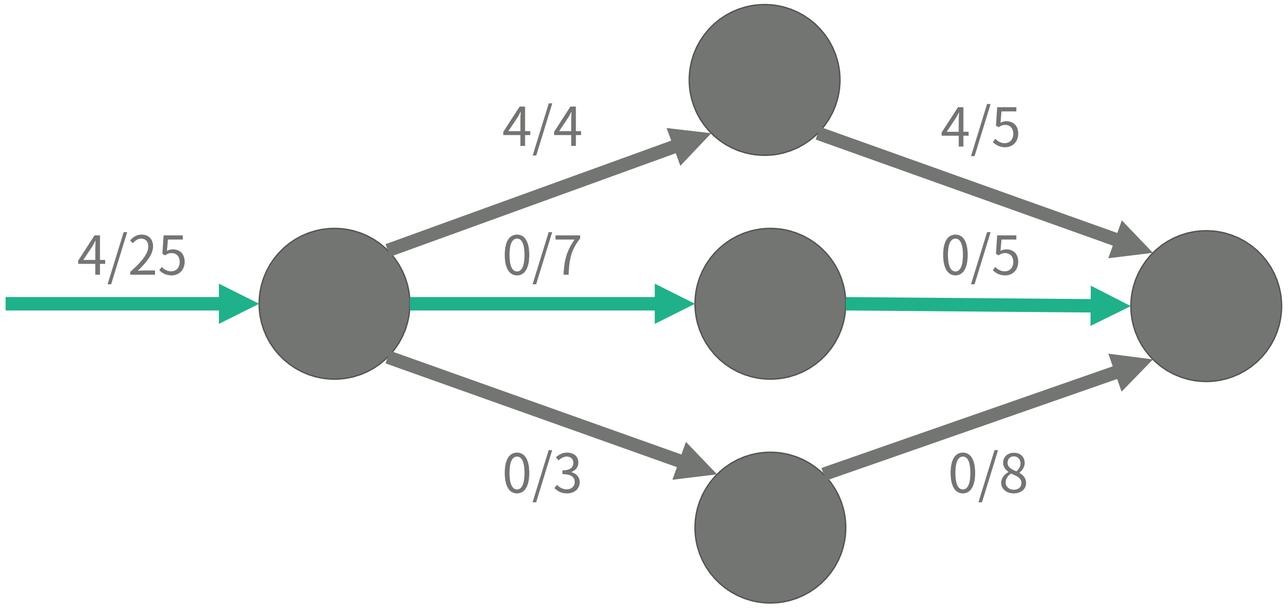
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

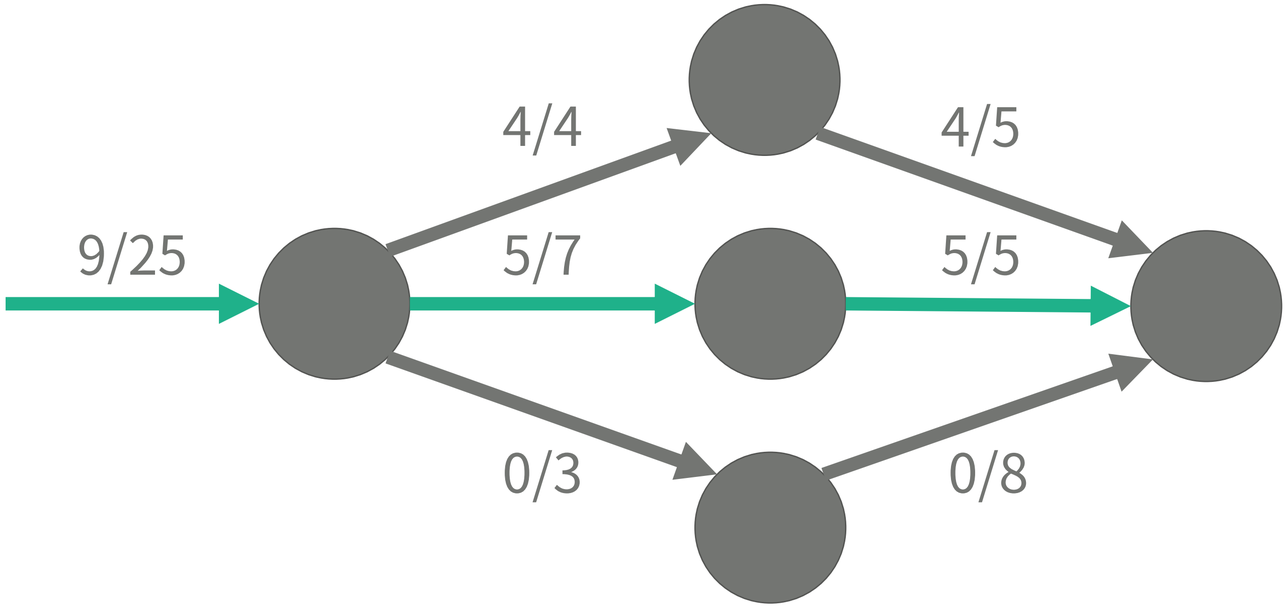
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

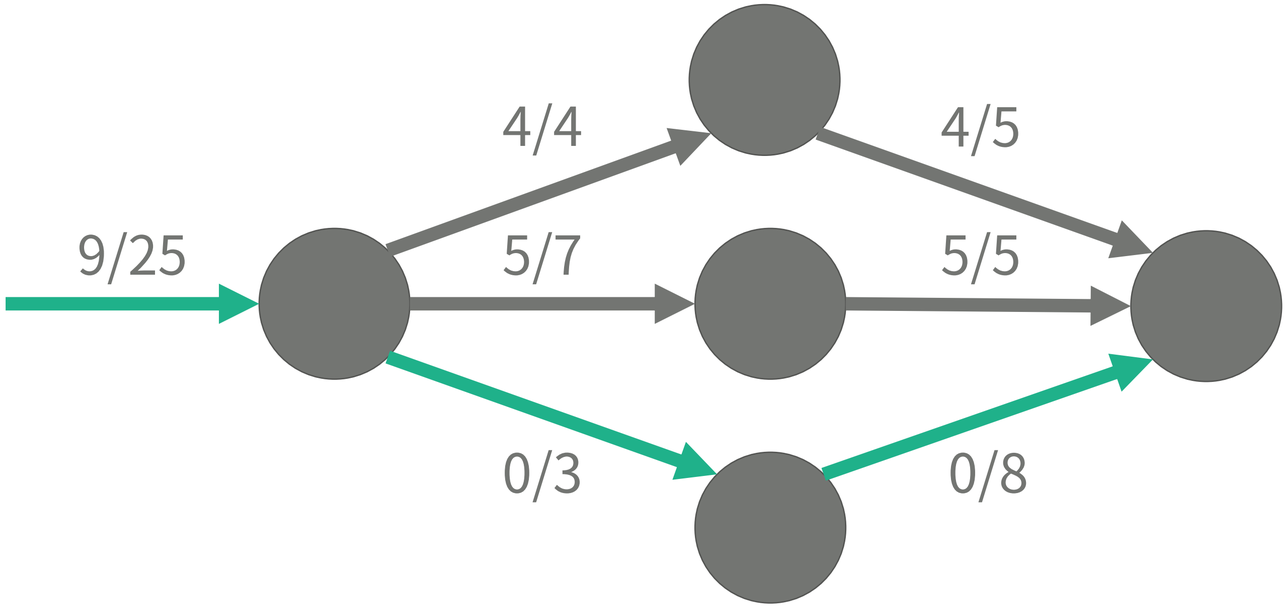
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

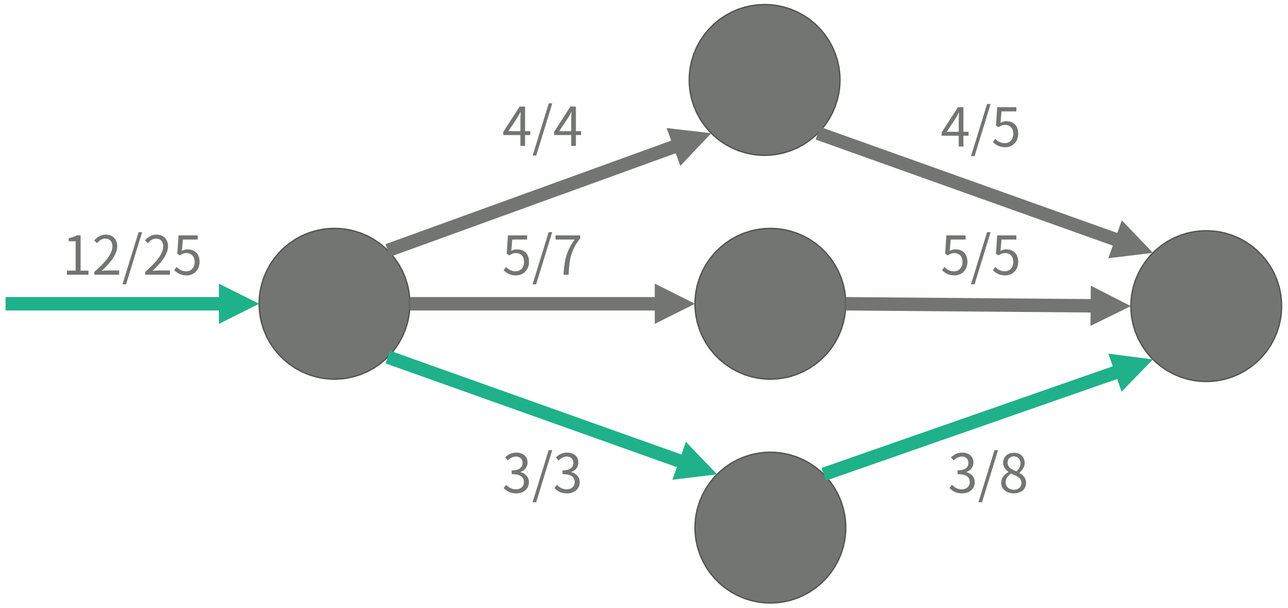
- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

Maximum Flow

- DFS를 이용해 Blocking Flow를 흘려준다.



Dinic’s Algorithm

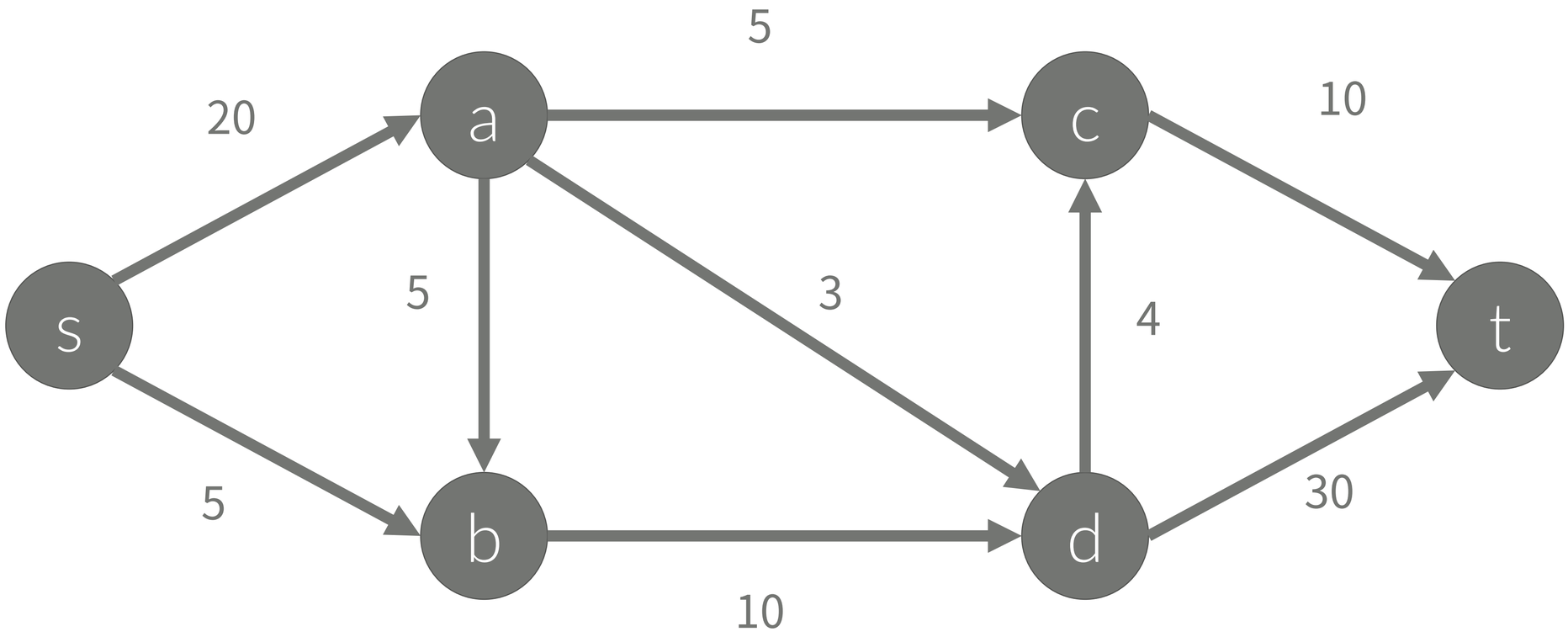
Maximum Flow

1. Level Graph를 만든다.
2. s → t로 가는 Augmenting Path를 찾는다.
3. Augmenting Path에서 최솟값을 찾는다. 이 때 정점을 b라고 한다.
4. b → t로 가는 Augmenting Path를 찾는다.
5. 더 이상 못 찾을때 까지 3과 4를 반복한다.

Dinic’s Algorithm

Maximum Flow

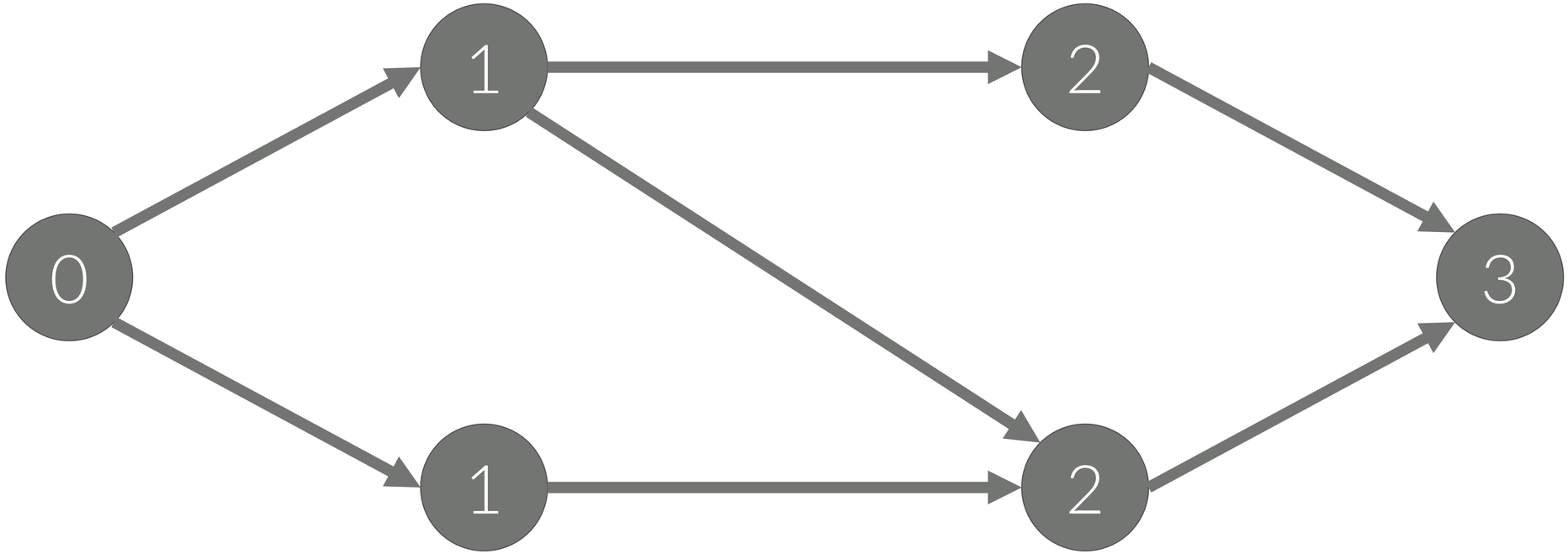
- Residual Graph



Dinic’s Algorithm

Maximum Flow

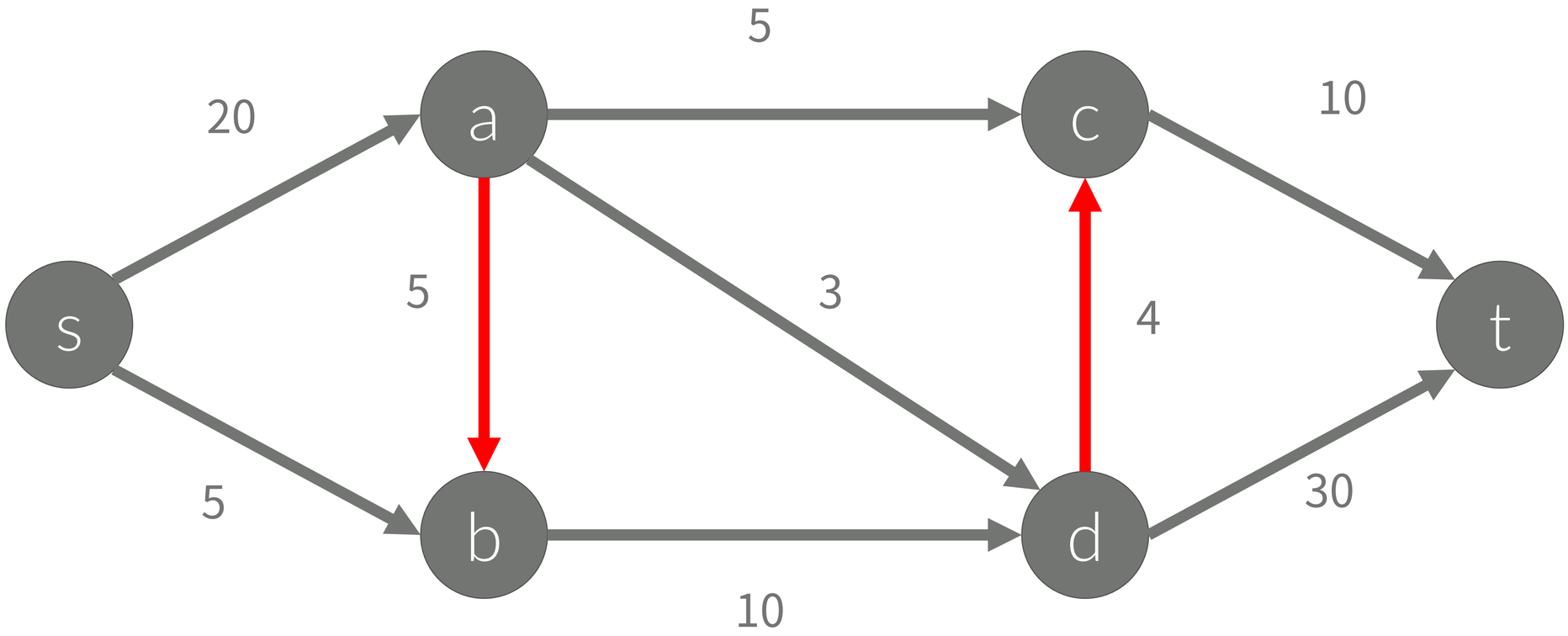
- Level Graph



Dinic’s Algorithm

Maximum Flow

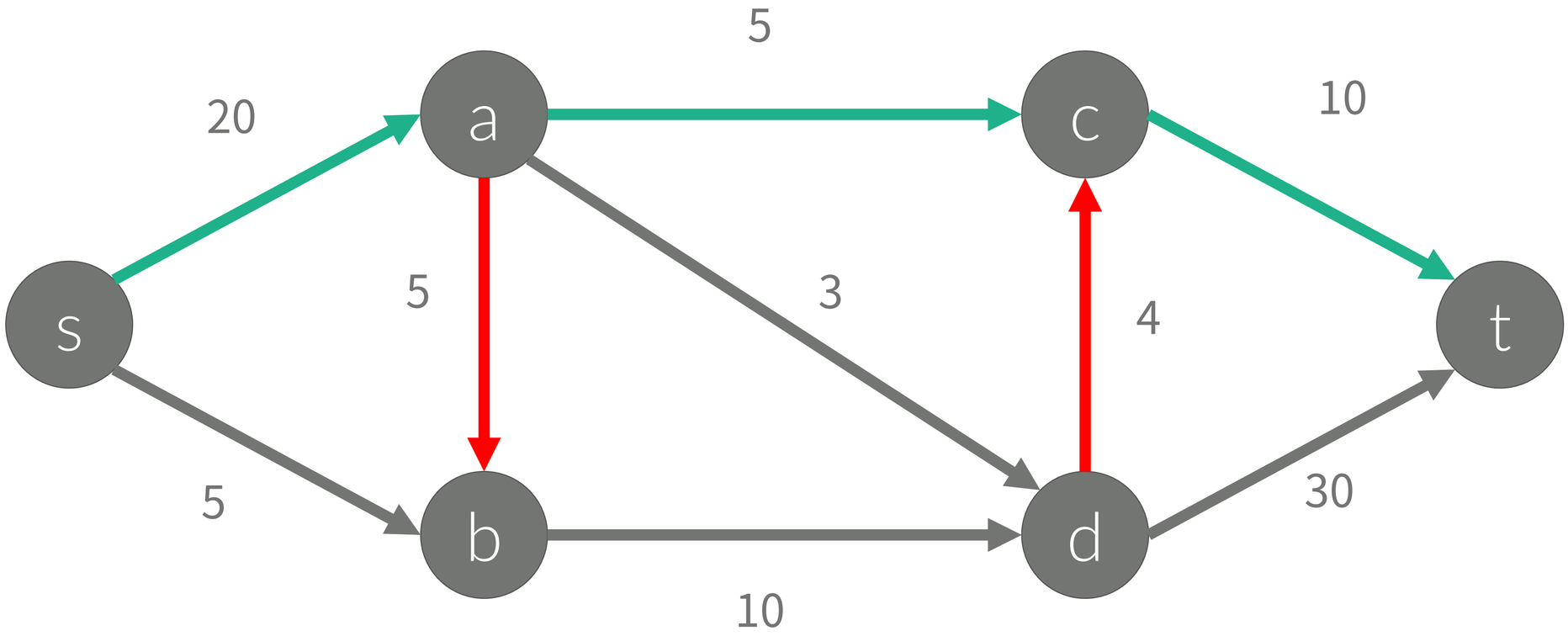
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

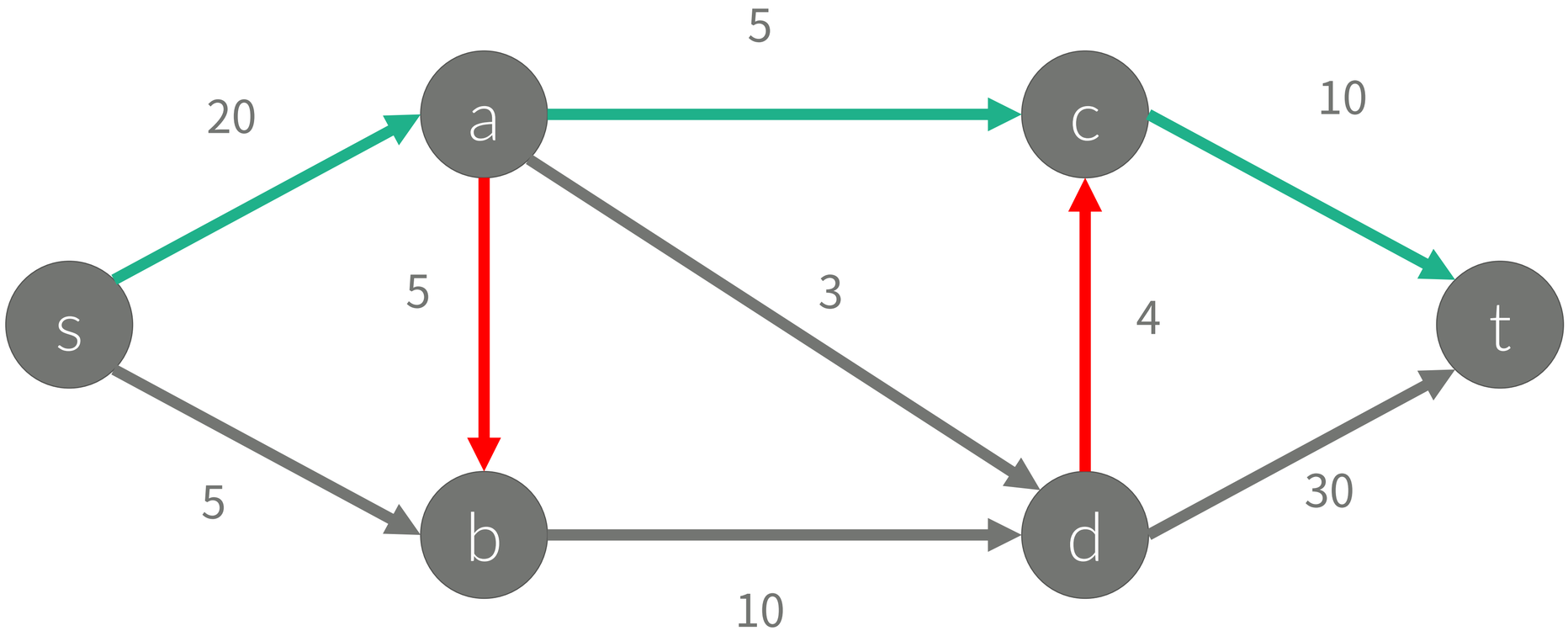
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

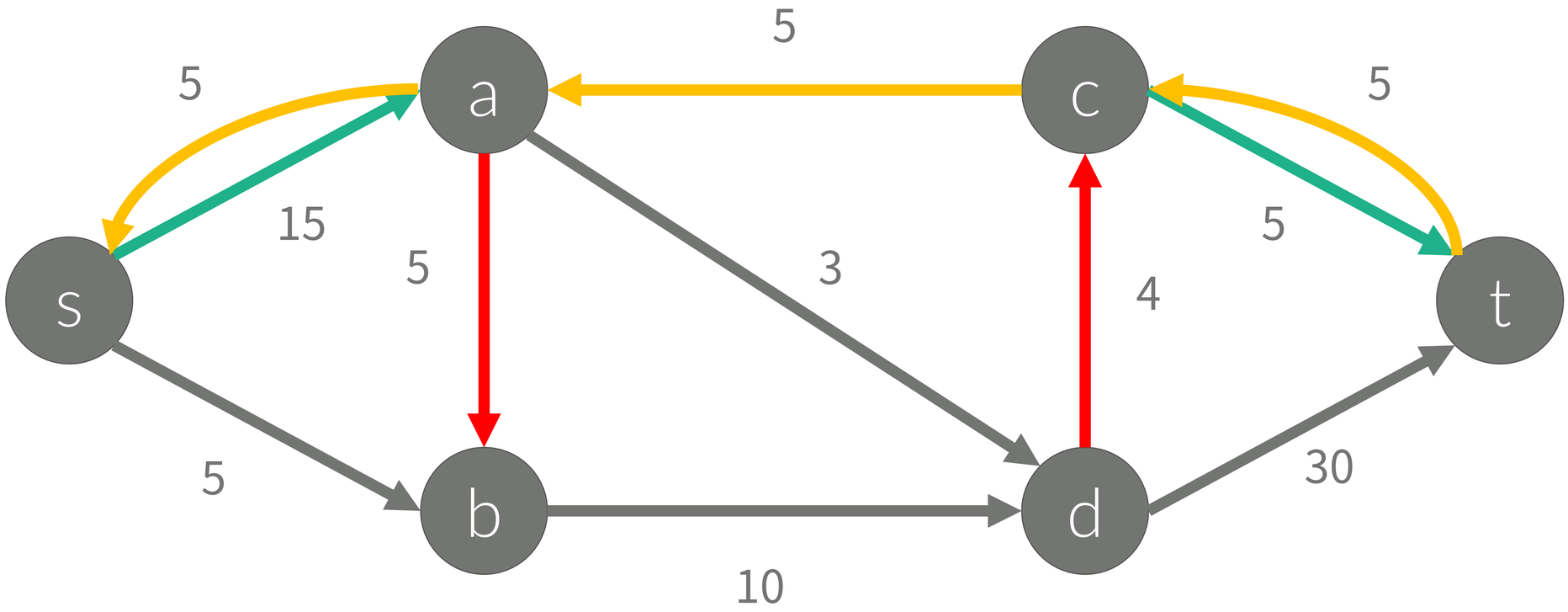
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

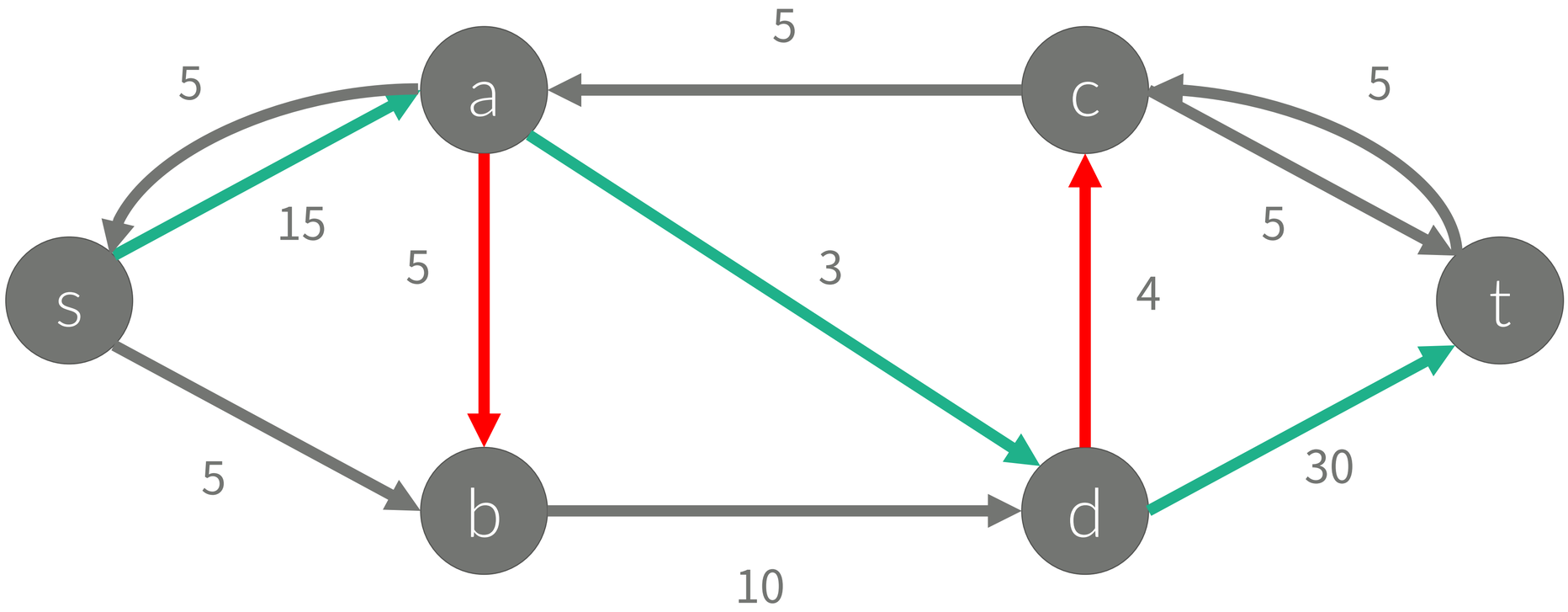
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

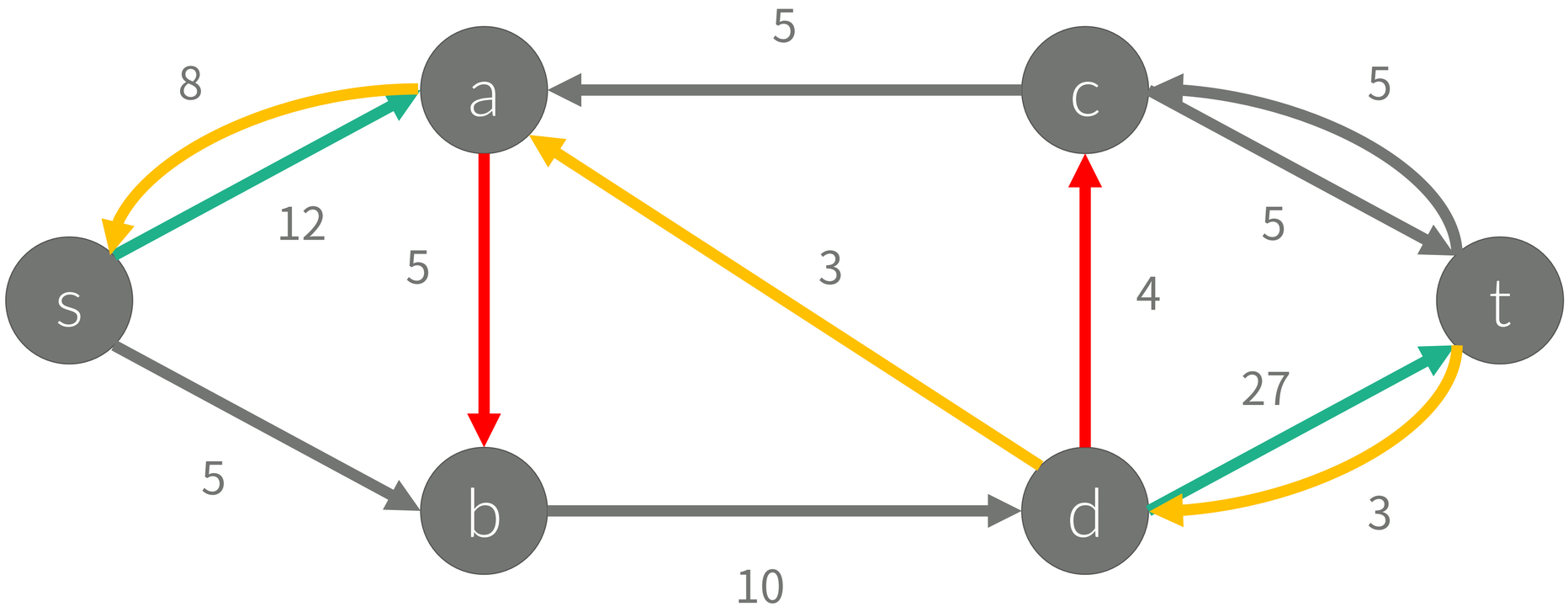
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

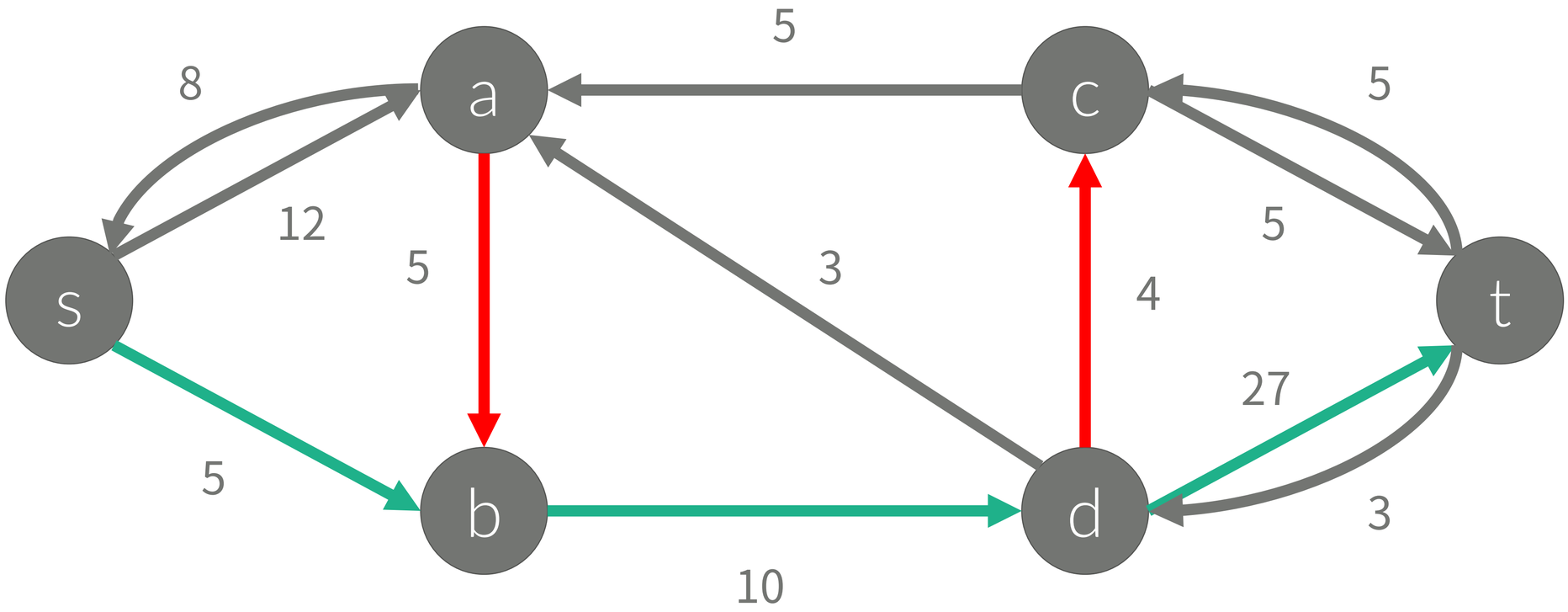
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

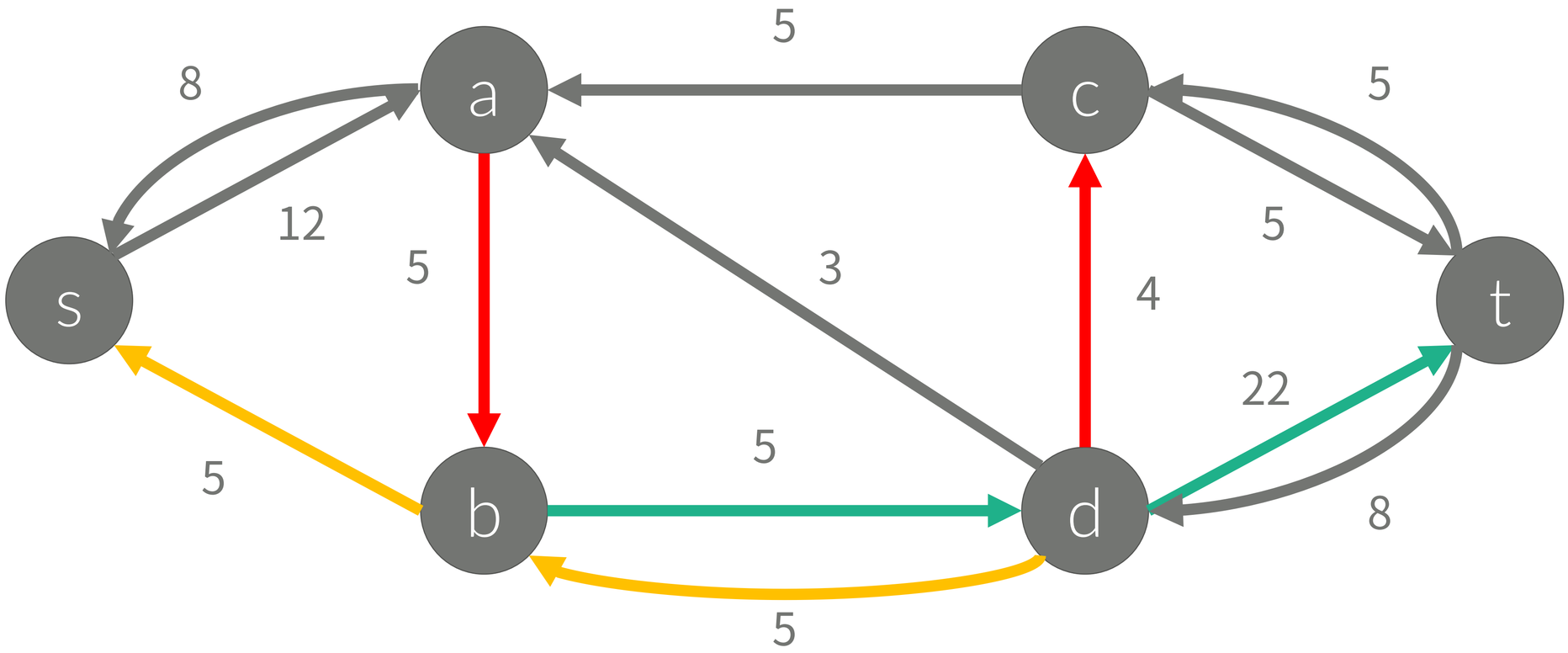
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

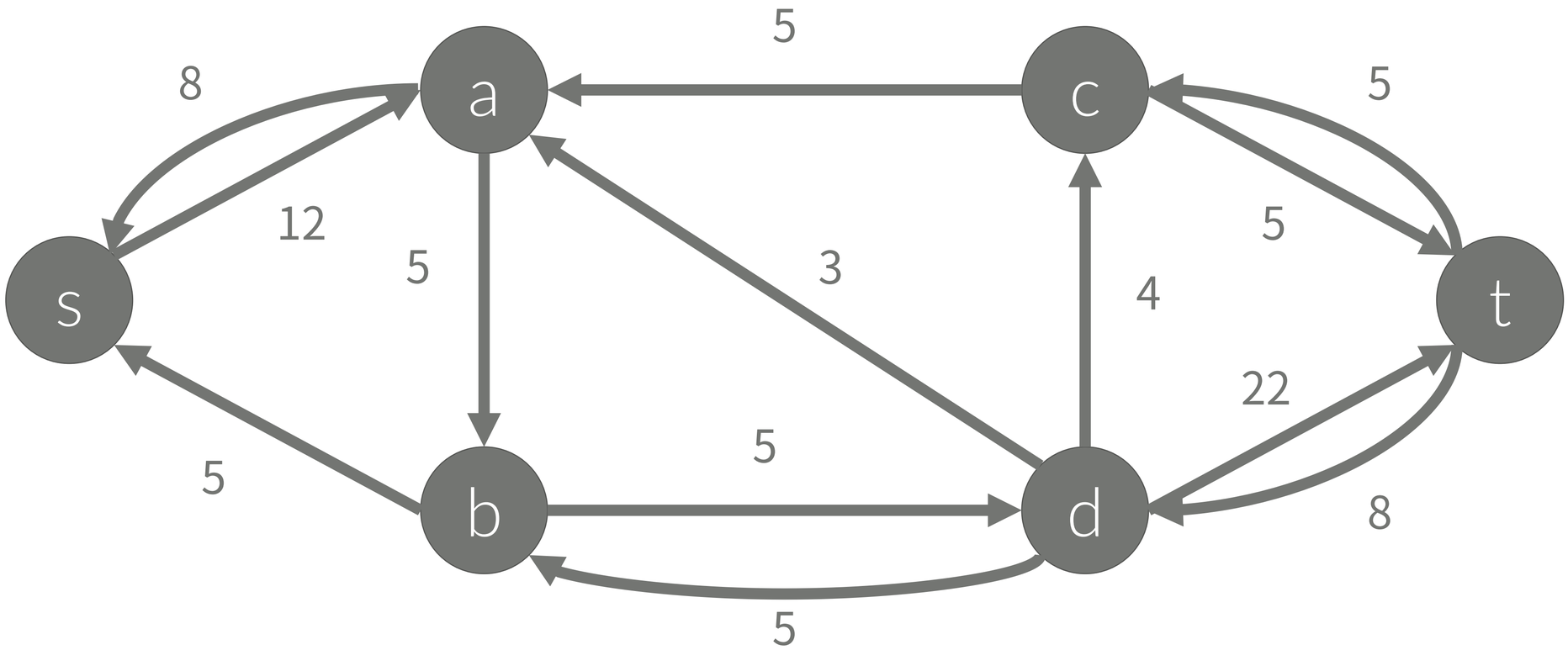
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

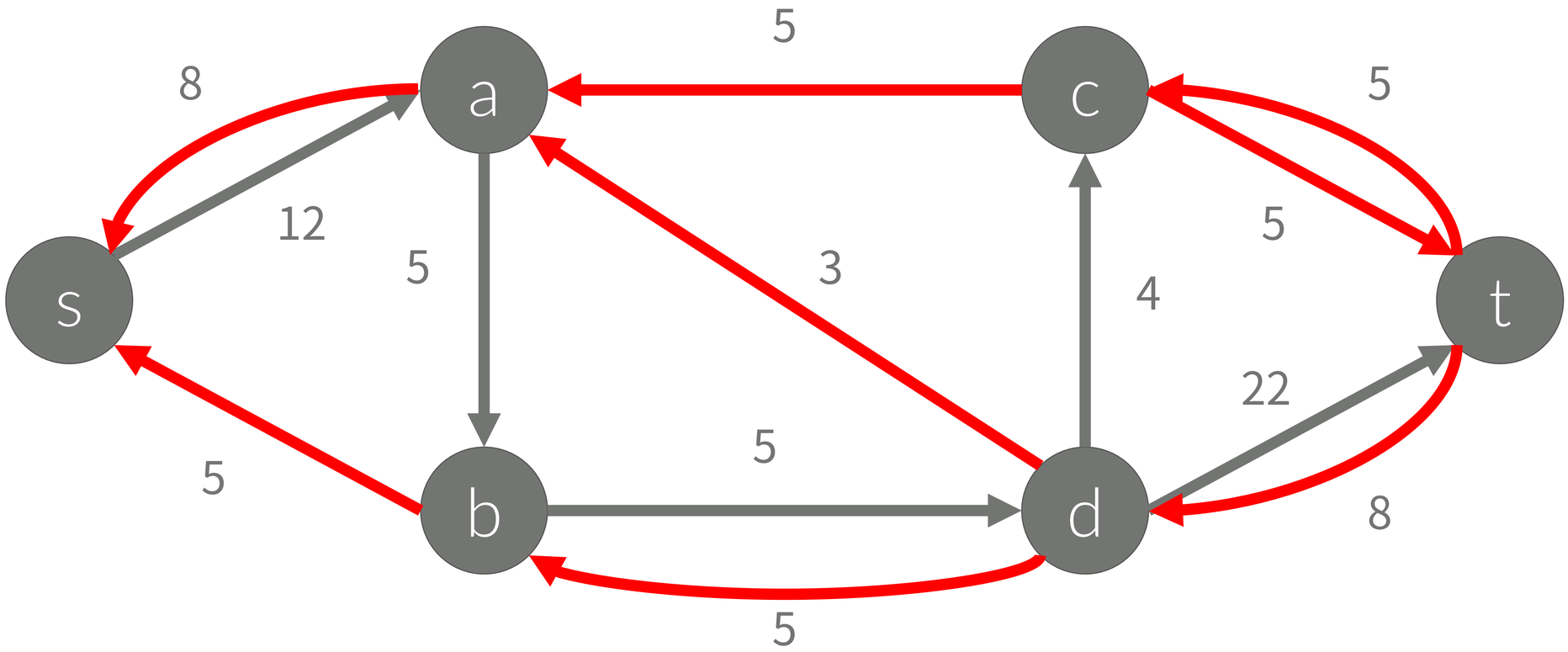
- Residual Graph



Dinic’s Algorithm

Maximum Flow

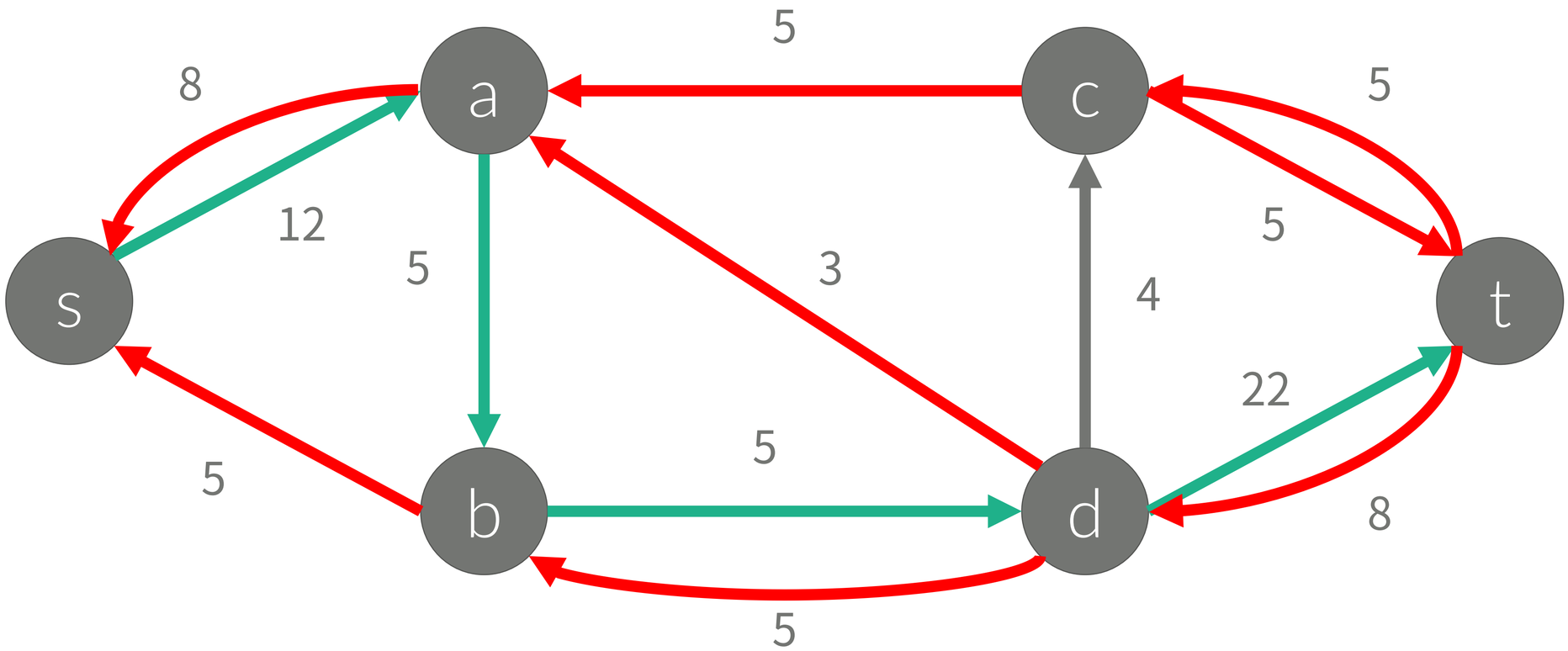
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

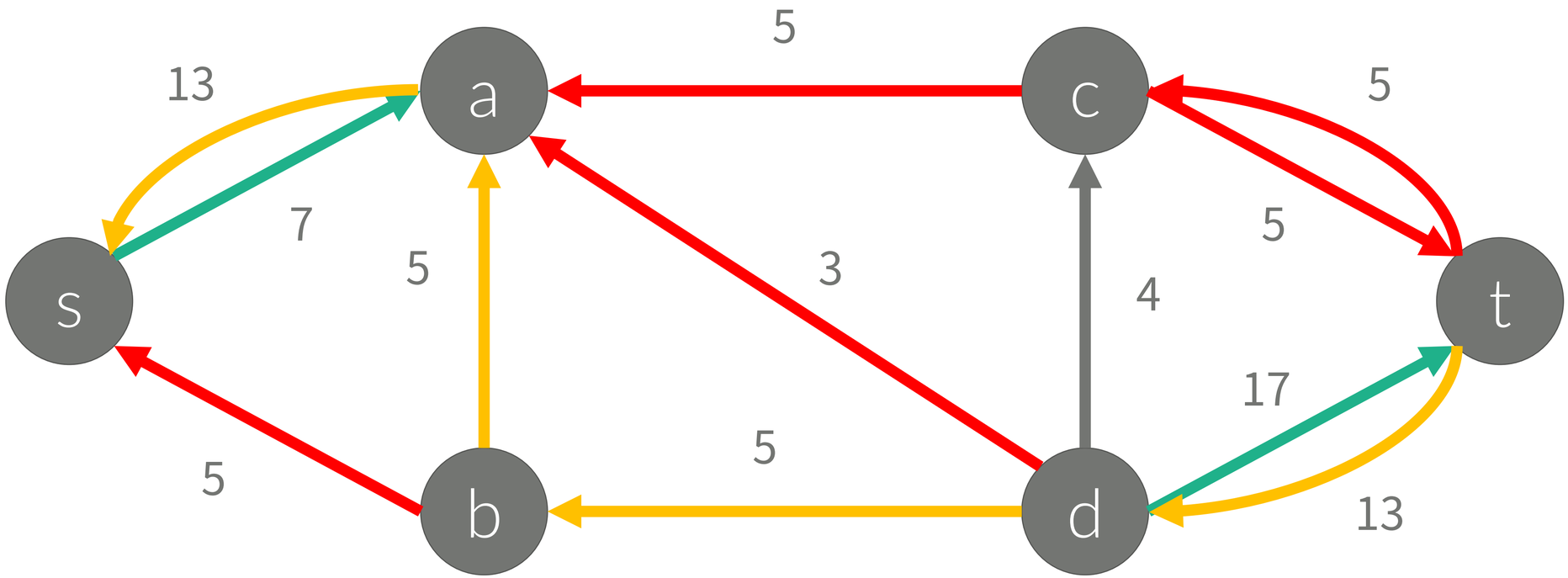
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

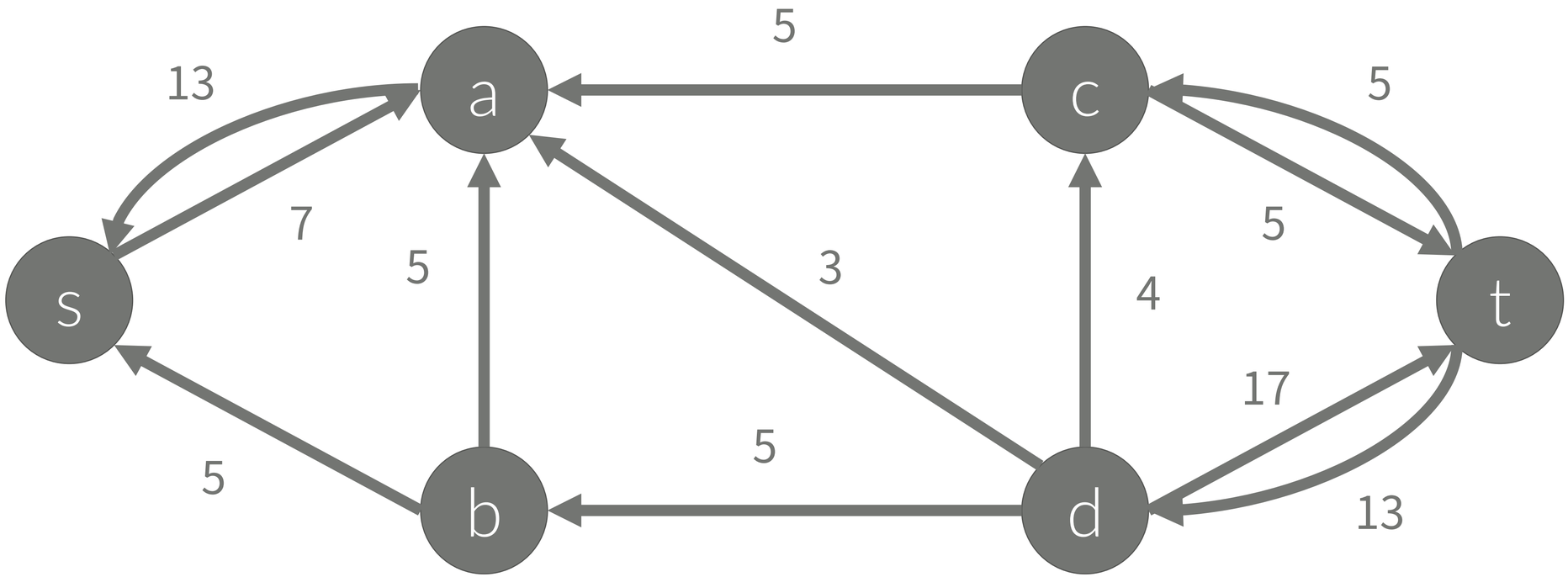
- Residual Graph +Level Graph
- 빨간 간선은 Level Graph에 없는 간선



Dinic’s Algorithm

Maximum Flow

- Residual Graph



도시 왕복하기 2

https://www.acmicpc.net/problem/2316

- 양방향 그래프 G가 주어진다. $3 \leq$ 정점의 수 ≤ 400, $1 \leq$ 간선의 수 $\leq 10{,}000$
- 1번 정점에서 2번 정점으로 가는 경로의 최대 개수를 구하는 문제
- 이 때, 한 경로에 포함된 정점을 다른 경로에서 사용할 수 없다.

도시 왕복하기 2

https://www.acmicpc.net/problem/2316

- 정점 V를 반으로 쪼개 Vin, Vout으로 만들고
- Vin -> Vout의 capacity를 1로 하면, 같은 정점은 1번만 방문하게 된다.

- 1번의 in이 소스, 2번의 out이 싱크인 경우 최대 유량과 정답이 같다.

도시 왕복하기 2

https://www.acmicpc.net/problem/2316

- 소스: http://codeplus.codes/e021f920a4df46b0a25b7164aa04a116