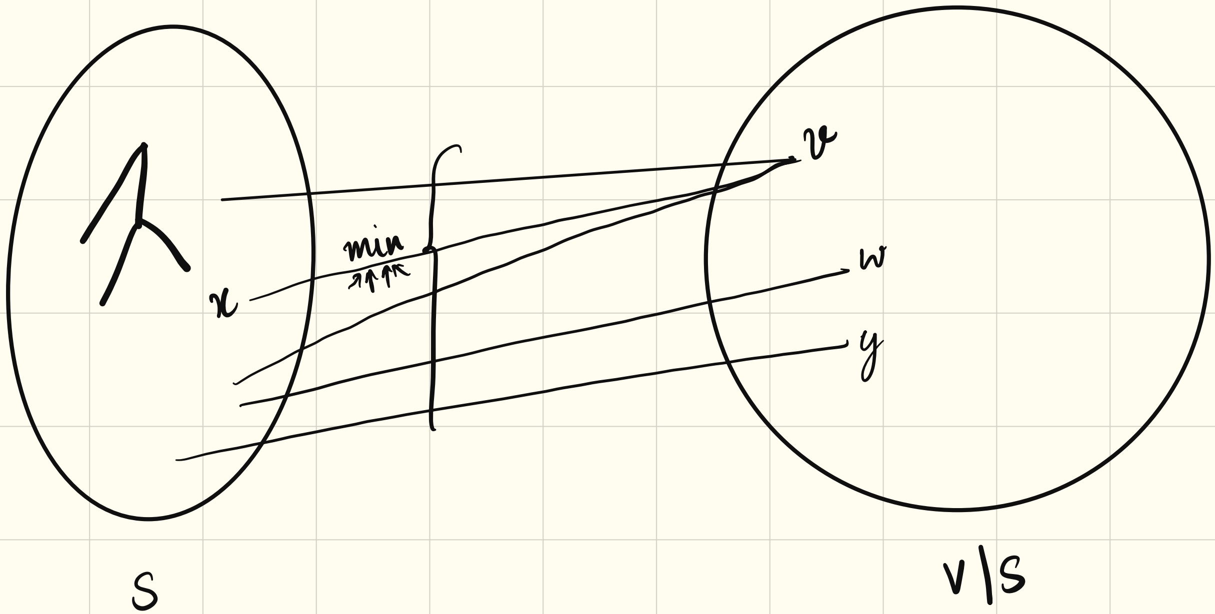
17 Apr 2025 - Algorithms - Week 15

Dijkstra's Algorithm

$d[v] :=$ minimum weight path from s to v



$$d[v] = \underbrace{d[x] + w(x, v)}_{\text{minimum}}$$

pred[v] = predecessor to a shortest $s \to v$ path using S

Maintain a tree on vertices in S.

Running time: See algorithm in PDF.

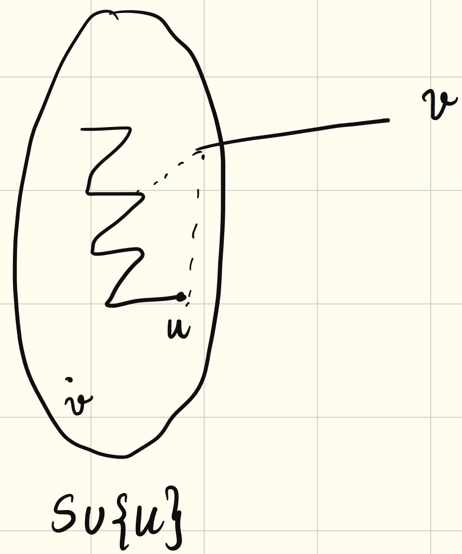
$$O(|V| \log |V|) \longrightarrow \text{extract-min}$$

$$+\ O(|E| \log |V|) \longrightarrow \text{relax}$$

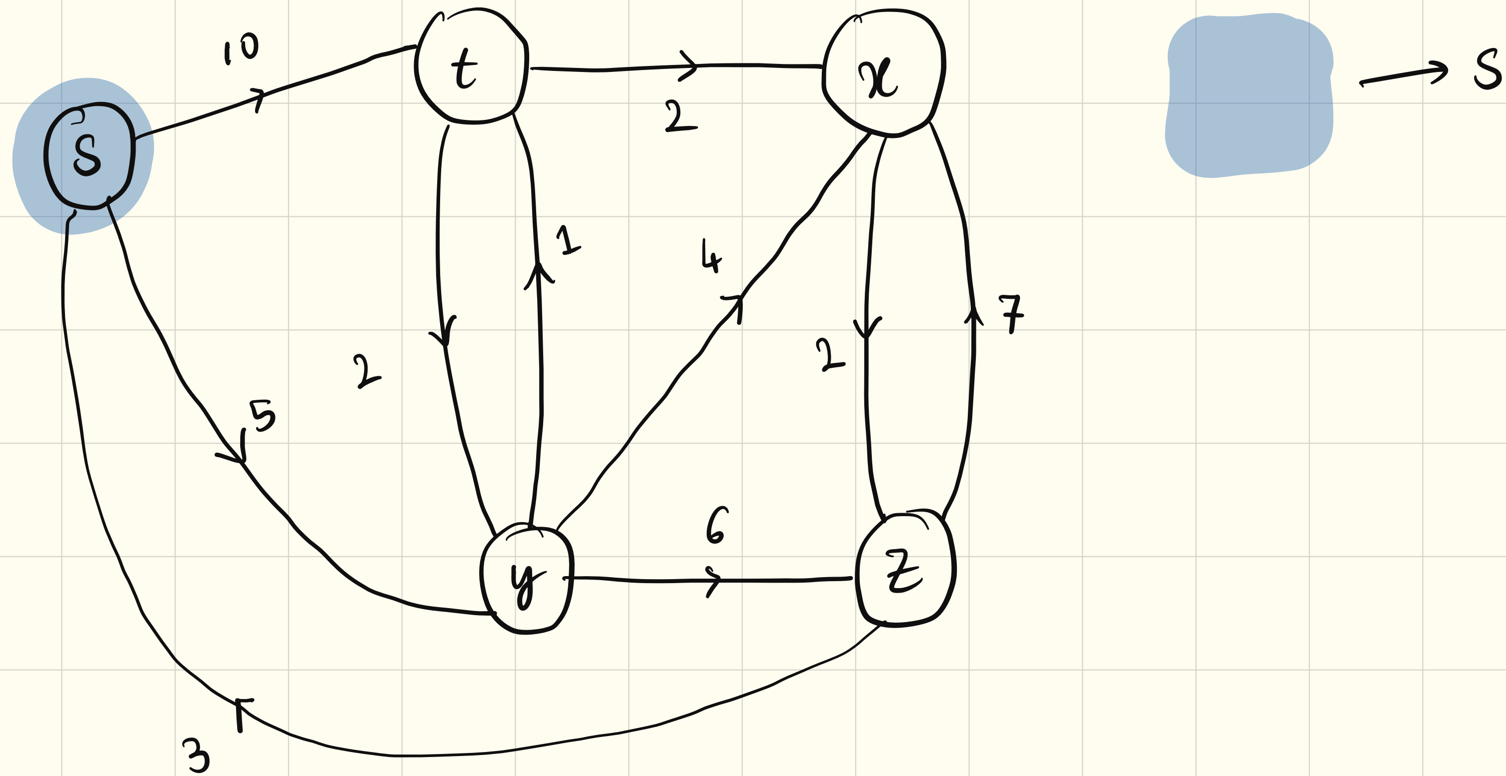
reduce key $\rightsquigarrow$ can be gotten rid of by using fibonacci heaps

$$\underbrace{\log |V| + \log(|V|-1) + \cdots + \log(1)}$$

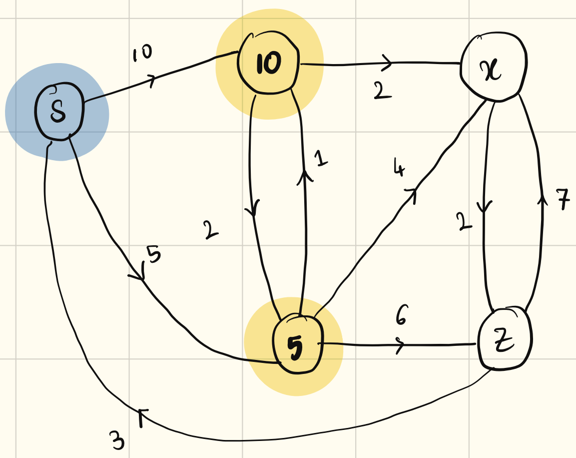
Claim: $d[v] = \delta(S \cup \{u\}, v)$

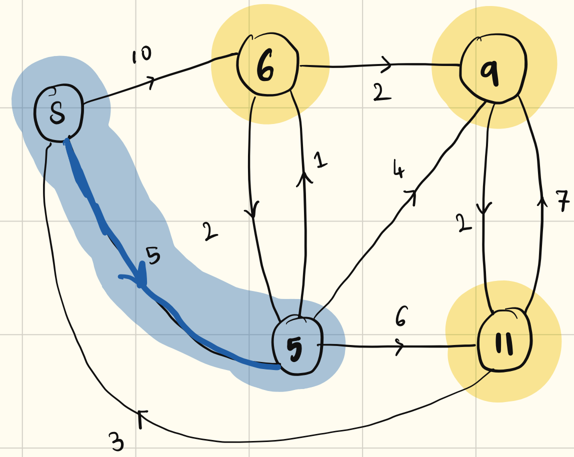


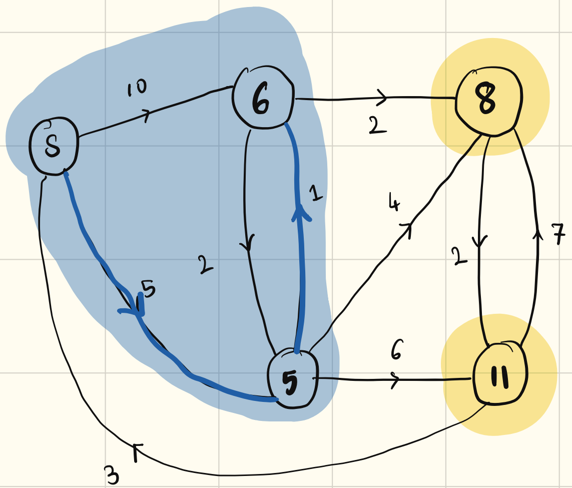
Proof: PDF

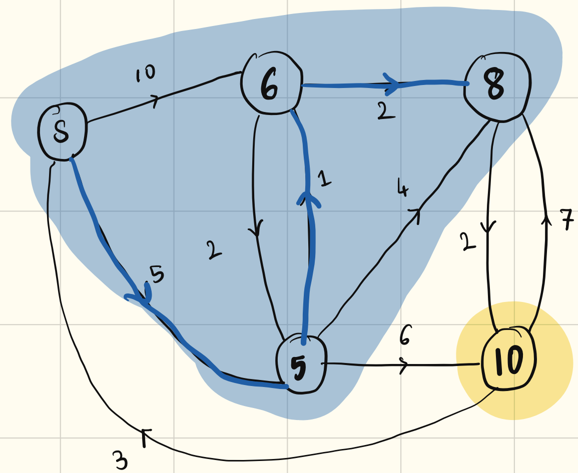


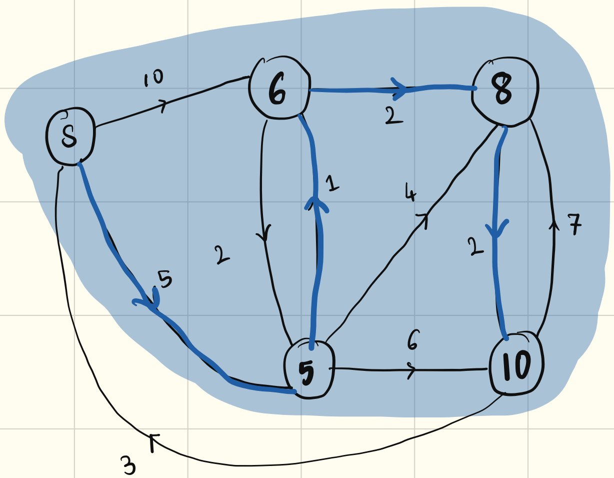
s
t
x
y
z
10
2
1
2
4
2
7
5
6
3
s


s
10
x
5
z
10
2
1
2
4
2
7
5
6
3


s
6
9
5
11
10
2
1
2
4
2
7
5
6
3


s
6
8
5
11
10
2
1
2
4
2
7
5
6
3


s
6
8
5
10
10
2
1
2
4
2
7
5
6
3



Negative weights

→ exchange rates and denominations.

Bellman - Ford Algorithm

→ works with negative weights also.

→ for $|V| - 1$ times:

for each edge (u, v)

Relax (u, v)

→ Time: $|V| \cdot |E|$

Initialisation : d

$$s \longrightarrow v_1 \longrightarrow v_2 \longrightarrow \cdots \longrightarrow v_k \qquad k \leq |V| - 1$$

$$v_k = v$$

relaxed in iteration 1

relaxed in ~~iteration 2~~ after v_1

relaxed

One more relaxation :

$$(u, v) \in E$$

$d[v] \downarrow \implies \exists$ a negative edge cycle of which this edge is a part.

All Pairs Shortest Path

Find $\delta(u,v)$ $\forall\ u,v \in V$

Apply single source shortest path algorithm $|V|$ times

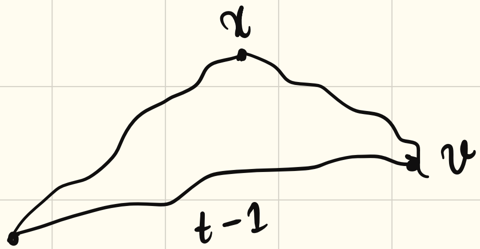
Dijkstra $\longrightarrow$ $O(|V|(|E| + |V|) \log |V|)$

Bellman-ford $\longrightarrow$ $O(|V|^2 |E|)$

Dynamic Programming 1

$f(u,v,t)$ = min weight path from u to v in $\leq t$ steps.

$$f(u,v,t) = \min f(u,v,t-1), \quad \min_{\substack{x \in V: \\ d[u,x] \\ < \infty}} \{f(u,x,t-1) + w(x,v)\}$$



$$f(u,v,t) = \min_{y \in Adj[u]} \{w(u,y) + f(y,v,t-1)\}$$

$|V|^3$ subproblems $\times$ $O(V)$

$$f(u, v, 100) = \min_{y} \{ f(u, y, 64) + f(y, v, 36) \}$$

32 32

16 16 . . .

split into powers of 2

~ divide and conquer

$f(u, v, t)$

↳ Compute this using $\log t$ values.

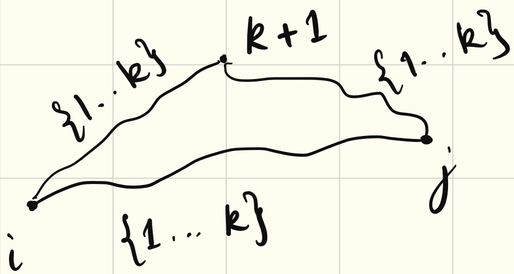
$O(|V|^3 \log |V|)$

↳ divide and conquer + DP

Dynamic Programming 2

$$V = \{ v_1, v_2, \dots, v_n \}$$

$$g(i, j, k) = \text{min cost of } i\text{-}j \text{ path using } \{1, 2, \dots, k\}$$



$$g(i, j, k+1) = \min\{ g(i, j, k),\ g(i, k+1, k) + g(k+1, j, k) \}$$

Time:

For each (i, j):

$O(|V|)$ comparisions $\rightarrow$ $k = 1$ to $|V|$
$- i, j$

For all $(i, j) \rightsquigarrow |V|^2$ choices

$O(|V|^3)$