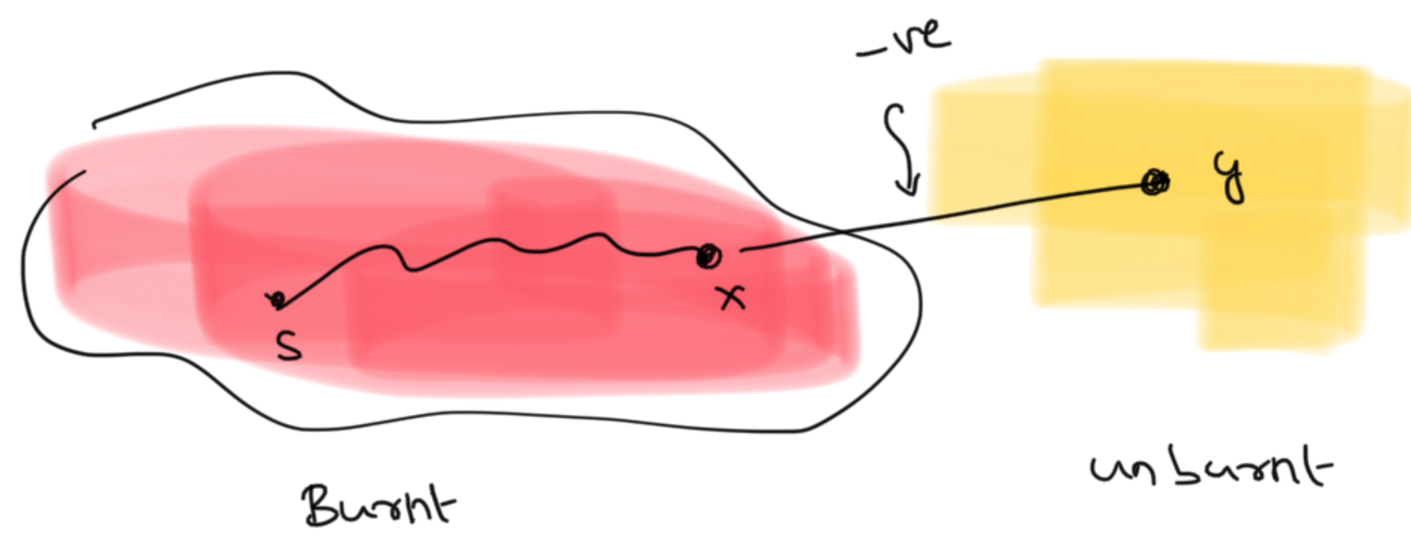
Bellman - Ford

Dijkstra:



→ each time you burn vertex, the distance computed <u>is</u> shortest path

→ But if negative edges allowed, not true anymore

as $d(s,x) \leq d(s,x) + x \xrightarrow{-ve} y$

↑
Burnt $d(s,x)$

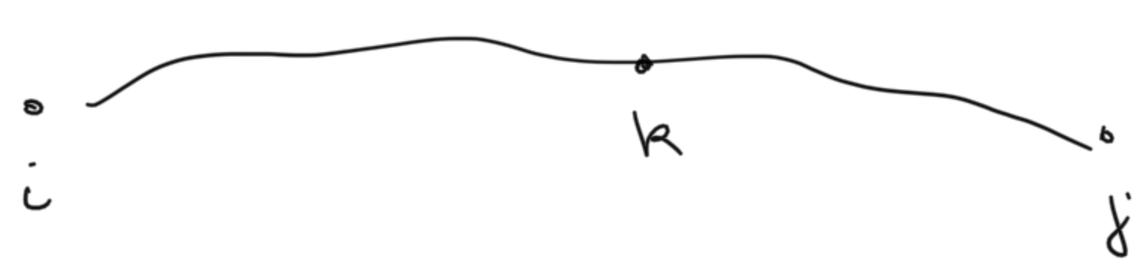
→ so Dijkstra fails if negative edge weights

However if no negative cycles, shortest path still makes sense. Need alternative algorithm

Observations

→ no visiting same vertex twice in shortest path

→ so if n vertices, shortest path $\leq n-1$

→ 

If $i - j$ shortest path $\Rightarrow$ i — k, k — j are also shortest paths

Dijkstra

when j burnt

U P D A T E

$$\text{Dist}(s,k) = \min\left[\text{Dist}(s,k),\ \text{Dist}(s,j) + \underbrace{j-k}_{\text{edge}}\right]$$

$\forall\ j \to k$

edges

→ Update operation $\forall$ nbhrs of j when j burnt

→ Distance dist(s,j) is guaranteed to be correct when you burn vertex

→ Dijkstra : chooses "smallest" unburnt vertex

→ Greedy strategy to cut down only to necc updates

When negative wts on edges

if you do update $j \to k$

$$\widetilde{\text{Dist}}(s,k) = \min\left[\widetilde{\text{Dist}}(s,k),\ \widetilde{\text{Dist}}(s,j) + \underbrace{j-k}_{\text{edge}}\right]$$

let Dist(s,j) be actual shortest dist.

guaranteed that

$$\text{Dist}(s,j) \leq \widetilde{\text{Dist}}(s,j) \quad \forall j$$

→ if $\text{Dist}(s,j) = \widetilde{\text{Dist}}(s,j)$

+ $s \rightsquigarrow j \xrightarrow{\text{edge}} k$

then $\text{Dist}(s,k) = \widetilde{\text{Dist}}(s,k)$ as well.

→ can't use greedy approach here...

??

want to exploit this.

Bellman-Ford

Ⓘ Initialize

$\rightarrow$ $\widetilde{Dist}(s, s) = 0$ [CORRECT]

$\widetilde{Dist}(s, j) = \infty \quad \forall\ j \neq s$

Ⓘⓘ Do update $\widetilde{Dist}(s, j)$ $\forall$ edges in graph

Do this $n-1$ times

where $n = |V|$

$\widetilde{Dist}^{k}(s, j) = k^{th}$ update.

Ⓘⓘⓘ $\widetilde{Dist}^{n-1}(s, j) =$ correct answer!

$O(|V||E|)$ $\leftarrow$ if adjacency list

$O(|V|^3)$ $\leftarrow$ if adjacency matrix.