$\boxed{\text{Lec 18}}$ More Dijkstra & Bellman - Ford

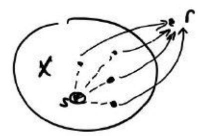
Priority First Search

```
search G s =
  X = {}        F = {s}
  init
  while |F| > 0:
    v = min_{x∈F} p(x)
    visit v
    X = X ∪ {v}
    F = N_G(X) \ X   ≡ (F\{v}) ∪ (N⁺_G(v)\X)
  return X
```

Dijkstra property

If no negative weight edges and define priority

$$p(v) = \min_{v \in X} (\delta(s,v) + w(v,r))$$



$$Y = \operatorname{argmin}_{v \in V \setminus X} p(v)$$

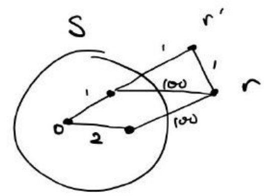
then $p(Y) = \delta(s, Y)$

Dijkstra's init:

$d(s) := 0 \qquad d(x) := \infty$ for $x \in V \setminus \{s\}$

$$p(v) = \min_{x \in X} (d(x) + w(x,v))$$

visit $d(v) = p(v)$



Ensures: $d(v) = \delta(s,v)$

<u>Impl</u>

```
dijkstraPQ G s =
  let loop X Q =
    case delmin Q of
      (NONE, _) ⇒ X
     |(SOME (d,v), Q') ⇒
       if (v, _) ∈ X then loop X Q'
       else let
```

like an augmented frontier with extra verts, or even duplicate verts due to diff dist to source by diff paths

priority queue Q

insert $Q \times (\mathbb{Z} \times \mathbb{N}) \to Q$

delmin $Q \to (\mathbb{Z}, \mathbb{N})$ option

```
        X' = X ∪ {(v,d)}
        relax (Q, (u,w)) = insert (Q, (d+w, u))
        Q'' = iterate relax Q' (N_G^+(v))
      in
        loop X' Q''
      end
  in
    loop {} (insert emptyQ (0,s))
  end
```

Dijkstra cost analysis

Observe parallelism maybe possible in ——— for equal weight batches and ——— with batch insertion enabled queue, but in general sequential.

every edge could cause insert

Operation	Number	Work
Q.delmin	m	$\lg n$ ($= \lg m$)
Q.insert	m	
T.find	m	
T.insert	n	
$N_G^+(v)$ (T.find)	n	
Calls to iterate	m	$O(1)$
Total work	$O((m+n)\lg n)$	

With fancy pqueue, $O(m + n\lg n)$ possible

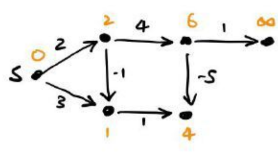
could be $O(1)$, but above still $O(\lg m)$

Belleman - Ford — min dist on arbitrary graphs, but less efficient

Ex. - convert from other probs lead to graph with neg edge
- find best way to convert currency when we want max product, take neg log and find min path, so we could get negatives

Intuition : $\delta^k(s,v)$ = shortest path $s \to v$ with max of k edges

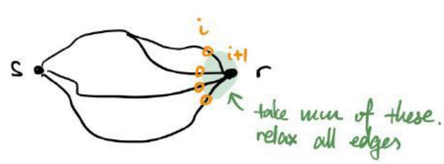
given $\delta^i(s,v)$ for all v get $\delta^{i+1}(s,v)$ by trying all next edge ← parallel!



$\delta^2(s,\cdot)$

get global : find δ^{n-1} *

* if not done at $i \geq n$ then we got neg cycle



```
BF G = (V,E) s =
  let loop (D: (V, ℝ) table) k =        just to prevent D[s] → ∞
    let D' = { v ↦ min ( D[v] ,   min          D[v] + w(u,v))  : v ∈ V }
                                u∈N⁻_G(v)
    in
      if (k = |V|) then NONE  ← neg weight cycle
      else if D = D' then SOME D
      else loop D' (k+1)
  in
    loop {s ↦ 0} ∪ {v ↦ ∞ | v ∈ V\{s}} 0
  end
```

$W(n,m) = O(mn)$ $\quad\quad$ $S(n,m) = O(n \lg m) = O(n \lg n)$