SHORTEST PATHS (SINGLE-SOURCE)

Tuesday, October 30, 2018 6:10 PM

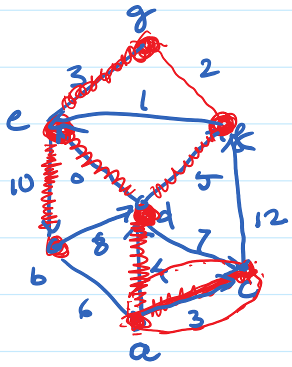
WHERE $w(e) \geq 0$ for every edge e

INPUT : a weighted digraph $G = (V, E, w)$ and a source vertex s

OUTPUT: shortest distance from s to v for all vertices v in G.

Dijkstra's algorithm is a greedy solution to this problem

Ex :



all weights are nonnegative

source vertex a

	a	b	c	d	e	f	g
d	0	14	3	4	4	9	7
parent		e	a	a	d	d	e

CLAIM : let x be the vertex such that $w(a, x)$ is smallest.

$d(x) = w(a, x)$

Proof :

Suppose $a \overset{w_1}{-} y_1 \overset{w_2}{-} y_2 - \cdots \overset{w_k}{-} x$ is a shorter path than $a \overset{w^*}{-} x$

$\underline{w_1 + w_2 + \cdots + w_k} \geq w^* + w_2 + \cdots + w_k$ (due to choice of w^*)

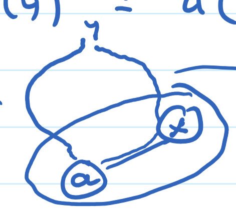
$\geq w^* + 0 + 0 + \cdots + 0$

$= w^*$

CLAIM : Suppose $d'(y)$ is the shortest distance from the contracted vertex $a\text{-}x$

Then $d'(y) = d(y)$ or $d'(y) + d(x) = d(y)$

$d'(y) = d(y)$



$d'(y) + d(x) = d(y)$

EFFICIENT IMPLEMENTATION OF DIJKSTRA'S ALGORITHM USING HEAP

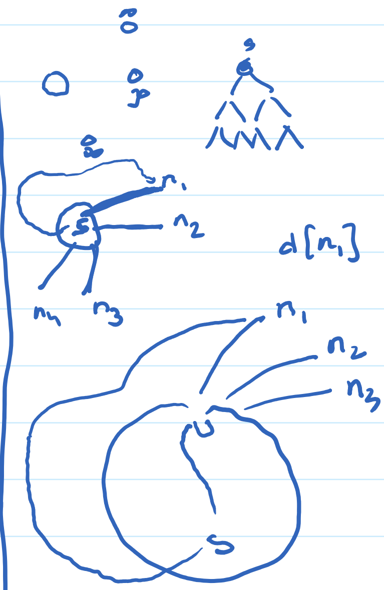
Dijkstra(V, E, W, s)

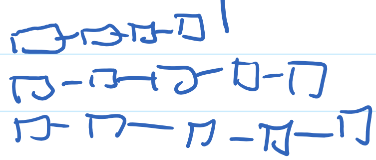
{

O( ) for each vertex v in V

```
Θ(n) {  for each vertex v in V
        {    v.d = ∞;
             v.parent = null;
        }

Θ(1) {  s.d = 0;
        T.V = {s}     // T is the shortest path tree
        T.E = { }
Θ(n) {  H = BuildHeap(V); // min heap based on .d
        for (i=1; i ≤ |V|; ++i)
        {
n           u = H.extract_min(); // O(lg n)   } O(n lg n)
            T.V += {u}
            T.E += {(u.parent, u)}  // O(1)   } O(n)
O(n+m)      for each v in Adj[u]
            {  if v ∈ H && v.d > u.d + w(u,v)
               {   v.d = u.d + w(u,v)
                   v.parent = u
                   H.decrease_Key(v, v.d); //  O((n+m) lgn)
               }
            }
        }
        return T;
}
```





Total cost: $O(n\lg n + (n+\underline{\underline{m}})\lg n)$

$= \boxed{O(m \lg n)}$