Dijkstra's Algorithm

Input: A simple connected undirected weighted graph *G* with nonnegative edge weights, determined by a weight function *wt* (*x,y*), and a starting vertex *s* of *G*.

Output: Array A of distances d(s,v) from s to v, for each v in V, so A[v] = d(s,v) for each v
Aux Output: Array B with property that B[v] is a shortest path from s to v.

The Algorithm:

A [*s*] $\leftarrow$ 0. *B* [*s*] $\leftarrow$ empty path (empty set)
X $\leftarrow$ {s} //Basis step

while X $\neq$ V **do**
{ POOL $\leftarrow$ {(v,w) $\in$ E | v $\in$ X and w $\notin$ X} }
(v′,w′) $\leftarrow$ search POOL for edge (v,w) for which A[v] + wt(v,w) is minimal
add w′ to X
A[w′] $\leftarrow$ A[v′] + wt(v′,w′)
B[w′] $\leftarrow$ B[v′] U {(v′,w′)}

Correctness

- Loop Invariant: I(i) is the following statement: (where i means iteration #i)

  (1) $|X| = i + 1$

  (2) $A[v] = d(s,v)$ for all $v \in X$

Dijkstra – Correctness (2)

Verification of I(i) for all iterations i = 1,2 … n-1.

Base case i = 1 , it is obvious that I(1) is true.

Induction Step: We assume I(i) is true, so |X|= i + 1 and A[v] = d(s,v) all v in X.

- Iteration i+1 causes one more vertex to be added to X, so |X| = i + 2
- During iteration i+1, algorithm locates (v′,w′) that has least greedy length among edges from X to V - X, and the algorithm sets A[w′] = A[v′]+d(v′,w′)
- To complete the induction, it suffices to show A[w′] is shortest path length from s to w′,i.e., A[w′] = d(s,w′)

Dijkstra – Correctness (3)

- Let q : s, …, y, z, …, w′ be a truly shortest path from s to w′, where z is first vertex in V - X encountered on the path q. Let L be the length of q. Let q_0 be the path s, …, y, z; we denote its length L_0. Notice that $L_0 \leq L$ (since no edge has negative weight). We will actually show that $A[w'] \leq L_0$, and this will finish the induction step.

- Notice that the sum of edge weights in q_0 from s to y is the true distance d(s,y) from s to y because q is a shortest path from s to w′ (if we could find a shorter path from s to y, we could also find a shorter path from s to w′). Therefore, by the induction hypothesis,

  $$L_0 = \text{length of } q_0 = d(s,y) + wt(y,z) = A[y] + wt(y,z).$$

- Recall from the previous slide that the algorithm so far has already defined $A[w'] = A[v'] + wt(v',w')$ and that this is the smallest sum of the form $A[u] + wt(u,w)$, for u in X and w not in X.

- It follows that $A[v'] + wt(v',w') \leq A[y] + wt(y,z)$ and so $A[w'] \leq L_0$. This completes the induction and proof of correctness.