CE2101/ CZ2101: Algorithm Design and Analysis

Week 5: Review Lecture

Ke Yiping, Kelly

Content

- Dijkstra's Correctness
- Prim's Algorithm and Its Correctness

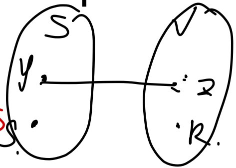
Dijkstra's Correctness

Lemma 1: The sub-path of a shortest path is also shortest.

min: $d(y)+W(y,z)$

- Prove by contradiction. $\forall y \in S,\ z \in V-S$

usage: already have a shortest path, extend on it.



Theorem D1: Assume that the shortest distances in S are correctly computed. The greedy step that chooses

for any y, any z.

$$\min\{d[y] + W(y, z) \mid \forall y \in S,\ z \in (V - S)\},$$

produces the shortest distance for z.

NANYANG TECHNOLOGICAL UNIVERSITY

Dijkstra's Correctness

Proof of Theorem D1 (continued)

Let P be a shortest path from **s** to **y** followed by edge (y, z)

Let W(P) = the distance travelled along P

Let P' = any shortest path different from P, i.e., P' = s, z1, …, zk, …, z

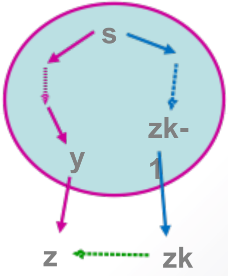
Assume that zk is the first vertex in P' not in set S.

W(P) = d[y] + W(y, z)

W(P') = d[zk–1] + W(zk–1, zk) + distance from zk to z

Note that: d[zk–1] + W(zk–1, zk) □ d[y] + W(y, z)

Since distance from zk to z is non-negative,
therefore, **W(P) <= W(P')**.



Dijkstra’s Correctness

Theorem D2: Dijkstra’s algorithm computes the shortest distances for every vertex correctly.

- Prove by mathematical induction on the size of S.
- The inductive step applies Theorem D1.

NANYANG TECHNOLOGICAL UNIVERSITY

Theorem D2 and Proof

Theorem D2: Given a directed weighted graph G with nonnegative weights and a source vertex s, Dijkstra's algorithm computes the shortest distance from s to each vertex of G that is reachable from s.

Proof (By induction):
We will show by induction that as each vertex v is added into set S, d[v] is the shortest distance from s to v.

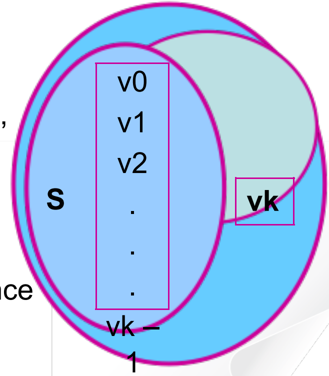
Basis: (Base Case)
The algorithm assigns d[s] to zero when the source vertex s is added to S. So d[s] is the shortest distance from s to s when S has the first vertex in it.

Theorem D2 and Proof (Continued)

Inductive Hypothesis: ✓ level 1: Proof by contradiction

Assume the theorem is true when S has **k** vertices. That is, assume v0, v1, v2, …, vk–1 are added where d[v1], d[v2]… are the shortest distances.

When vk is chosen by Dijkstra's algorithm, it means an edge (vi, vk), where i ∈ {0, 1, 2, …, k – 1}, is chosen to minimise d[vi] + W(vi, vk) among all edges with one vertex in S and one vertex not in S.

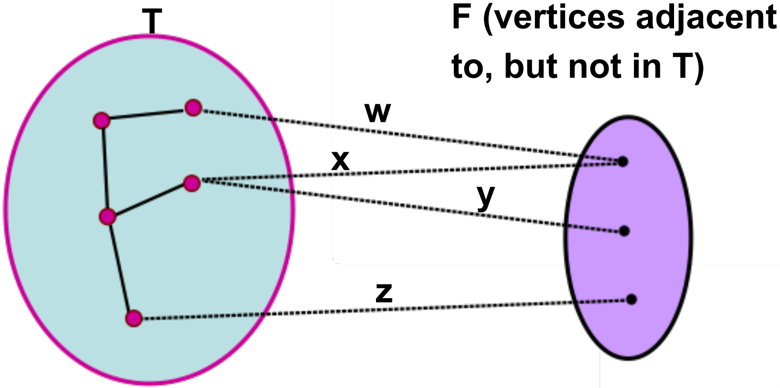


By Theorem D1, d[vk] is the shortest distance from source to vk. So the theorem is true when S has **k + 1** vertices.

Prim's Algorithm

- To compute a minimum spanning tree on an undirected, weighted graph.
- The input graph must be connected.

Main Idea of Prim's Algorithm

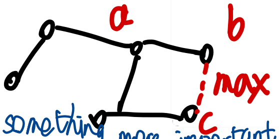


Choose min(w, x, y, z)

Prim's Correctness

Need this to help prove correctness of Prim's Algo.

MST property: If we add any edge that is in G but not in a spanning tree *T*, this edge will create a cycle in *T* and carries the max weight in this cycle.



$(b, c) \in E(G)$

引理: used to prove something more important

Lemma 1: If two spanning trees in G both have the MST property, then they have the same total weight.

- Prove by mathematical induction on the number of differing edges.

Prim's Correctness

Theorem 1: MST = MST property

- Only if: prove by contradiction. (If MST doesn't have MST property)
- If: prove by "only if" result and Lemma 1.

(For any intermediate step. the sub-tree have MST properties)

Lemma 2: The tree constructed at every step of Prim's has the MST property in induced subgraph.

- Proof not required.

At last step. the tree would have MST property

→ Prove by Lemma 2 and Theorem 1 tree with MST property is MST

Theorem 2: Prim's correctly computes the MST.

Exercise

- Given a directed graph G with non-negative edge weights, Dijkstra's algorithm is used to compute a shortest path P from a source vertex s to a target vertex t. If the weight of every edge in G is increased by 10, does P remain the shortest path from s to t in the modified graph? If your answer is "yes", give a proof. Otherwise, give a counterexample. **[AY1920S1]**

No. Assume there are l edges $(e_1, e_2, e_3, \ldots e_l)$.

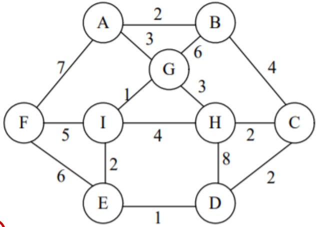
Original G: shortest P: $W(P) = \sum_{i=1}^{l} W(e_i)$

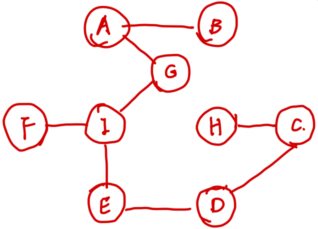
New Graph G': $W(P') = \sum_{i=1}^{l} W(e_i) + 10 = 10l + W(P)$

Counter example:

Exercise

- Execute Prim's algorithm on G in the figure starting at vertex A. Draw the minimum spanning tree obtained. **[AY1920S1]**





NANYANG TECHNOLOGICAL UNIVERSITY

Exercise

- Does Prim's algorithm work if some edges in the input graph have negative weights? If your answer is "yes", give a proof. Otherwise, give a counterexample. **[AY1819S1]**

Yes By theorem 1 and Lemma 2, can prove the correctness of Prim's . . .

In all prove, there's no special mentioning on the sign of weight.

need to point out to get full mark

No need to go into details.