Lecture 12 Shortest Paths

Tuesday, February 26, 2019

1. \( S \rightarrow t \) shortest path: find a path from \( S \) to \( t \) that has minimum length.

2. Single source shortest path: Given a source \( S \), find shortest path from \( S \) to all other vertices.

3. All pairs shortest path: find the shortest path between any pair of points.

- Dynamic Programming Structure
  - Let \( d[i,j] \) be the length of shortest path from \( S + i \) to \( S + j \).
    
    \[
    d[i,j] = \min_{(u,v) \in E} \{ d[i,u] + w(u,v) \}
    \]

    \[
    d[S] = 0 \quad d[S,a] = 5 \quad d[S,b] = 6 \quad d[S,c] = 7
    \]

    \[
    d[S,t] = \min \left\{ \begin{array}{l}
    d[S,a] + w(a,t) \\
    d[S,b] + w(b,t) \\
    d[S,c] + w(c,t)
    \end{array} \right\}
    \]

- Greedy fails

- Fixing the cycle problem
  
  \[
  d[i,v] = \min_{(u,v) \in E} \{ d[i,u] + w(u,v) \}
  \]
  
  \[
  d[i,u] < d[i,v] \]

- Dijkstra's Algorithm
- Dijkstra's Algorithm
  - mark a vertex $u$ to be visited if we know the shortest path from $s$ to $u$.
  - maintain $\text{dis}[u]$: shortest path from $s$ to $u$ that only uses visited vertices as intermediate points.

- every step, pick the vertex that has not been visited and has the smallest $\text{dis}[u]$.
  - claim: $\text{dis}[u] = d[\text{Cu}]$
  - mark $u$ as visited
  - update $\text{dis}[]$ accordingly.

- Proof of correctness for Dijkstra
  - Induction Hypothesis:
    at $i$-th iteration, know shortest path to $i$ visited vertices $\text{dis}[u]$ for any vertex $u$ is maintained correctly.
- Base Case.
  Initially, know shortest path to $s$, $s$ is only visited vertex
  \[ \text{dist}(u) = w(s,u) \text{ if } (s,u) \in E \]
  \[ \text{dist}(u) = +\infty \text{ if } (s,u) \notin E \]

- Induction: suppose IH is true for iteration $i$
  - Let $u$ be the vertex with smallest $\text{dist}(u)$ in this iteration
    - Claim: $d(u) = \text{dist}(u)$

- Proof: assume $d(u) < \text{dist}(u)$
  there exists another path from
  $\overline{s}$ to $u$ with shorter distance.
  By definition of $\text{dist}(u)$, this alternative
  path must have used some intermediate
  vertex that has not been visited.
  - Let $v$ be the first vertex on the path that is not visited.
    the length of path from $s \to u$ is at least $\text{dist}(v)$
    
    \[
    \text{total length of path} \geq \text{dist}(v) + d(u, v) \\
    \geq \text{dist}(v) \geq 0
    \]
    contradiction. There cannot be a shorter path.
    - still need to prove $\text{dist}(v)$ are maintained correctly

- Running time
  - Naive implementation: implement $\text{dist}[]$ as an array
    finding smallest element $O(n) \times n$
    update $\text{dist}[]$ $O(1) \times m$
total \quad O(n^2 + m) = O(n^2)

- use a binary heap for $\text{dis}[]$
  - finding smallest element \quad $O(\log n) 	imes n$
  - update $\text{dis}[]$ \quad $O(\log n) \times m$
  - total \quad $O(n\log n + m\log n) = O(m\log n)$ if graph is connected.

- use a Fibonacci heap for $\text{dis}[]$
  - finding smallest element \quad $O(\log n) \quad n$
  - update $\text{dis}[]$ \quad $O(1) \quad m$
  - (dis values can only decrease)
  - total \quad $O(n\log n + m)$