CSCI 323 Lecture 10 Note Wednesday, 03/09/2022

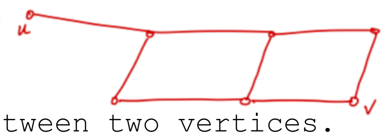
Announcements:
- The midterm will be on Wednesday, March 23, 2022.
- The Lecture Recordings will be available on the following YouTube Playlists Link: https://youtube.com/playlist?list=PLZaTmV9UMKlgYpo2cAiMaEWxqyvbiXDFd

Greedy Algorithm

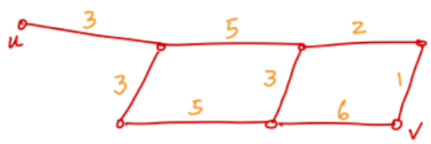
References:
Algorithm Design - Chapter 4.4, 4.5

Shortest Path – Dijkstra's Algorithm [4.4]

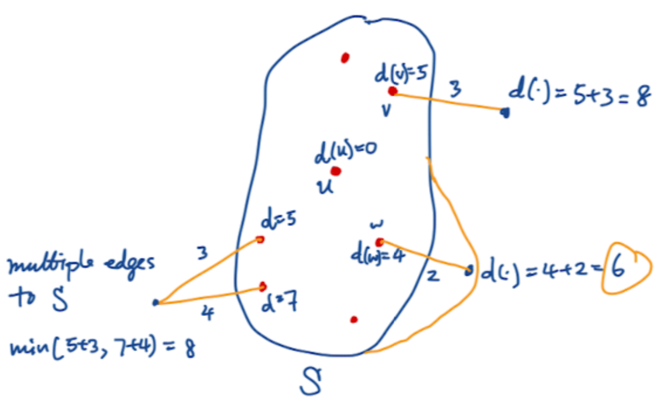
- Given a weighted (directed) graph $G = (V, E)$, where each edge $(u, v) \in E$ has a length/weight $l_{(u,v)}$.



- Goal to find the shortest path between two vertices.
    - o If all weights on the edges = 1, shortest = least number of edges.
    - o $BFS(u)$ solve the problem.
    - o What if weights are not all 1? (All weights are positive.)
    - o If the weights are negative, Dijkstra's algorithm cannot solve the problem.



- Dijkstra's algorithm from a source u, returns
    1) the length of the shortest path from u to any vertex v.
    2) the shortest path from u to v, for any v.
- Dijkstra's Algorithm:
    1) Maintains a set of explored nodes S. Initially $S = \{u\}$.
    2) Also maintains a function $d(v)$ for distance from u to v. Initially $d(u) = 0$, $\forall v \neq u, d(v) = \infty$.
    3) Adds vertices to set S one-by-one. Once a vertex v is added to S, its label $d(v)$ never changes and $d(v) = \textit{length of shortest path from u to v}$.

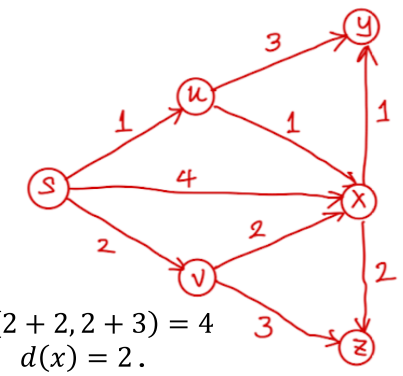


1) Consider vertices that have an edge to some vertex in S.
2) Calculate a label for each of these vertices.
3) Select the one with the smallest label, add that vertex to S in this step, recurse.

- Dijkstra's Algorithm(G,u):

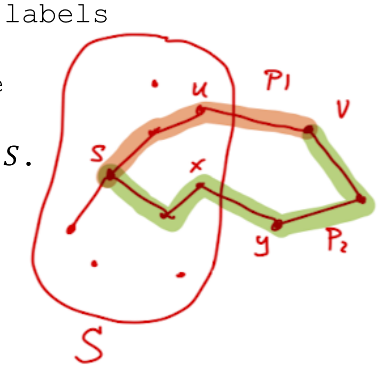
    Let S be the set of explored nodes
    For each $v \in S$ we store $d(v)$
    Initially $S = \{u\}$ and $d(u) = 0$.
    While $S \neq V$ (V is the set of all vertices)

Select a node $v \notin S$ with at least 1 edge from S
for which $d'(v) = \min_{\substack{(w,v)\in E,\\ w\in S}} \left(d(w) + l_{(w,v)}\right)$ is as small as possible.
Add v to S, set $d(v) = d'(v)$.
End while



- Example
  1) Initially $S = \{s\}$ and $d(s) = 0$.
     $d'(u) = 1, d'(x) = 4, d'(v) = 2,\ d(y) = d(z) = \infty$
  2) $S = \{s,u\}$ and $d(s) = 0,\ d(u) = 1$.
     $d'(y) = 4, d'(x) = \min(4,2) = 2, d'(v) = 2, d(z) = \infty$
  3) $S = \{s,u,x\}$ and $d(s) = 0,\ d(u) = 1,\ d(x) = 2$.
     $d'(y) = \min(1+3, 2+1) = 3, d'(v) = 2, d'(z) = \min(2+2, 2+3) = 4$
  4) $S = \{s,u,x,v\}$ and $d(s) = 0,\ d(u) = 1,\ d(v) = 2,\ d(x) = 2$.
     $d'(y) = 3, d'(z) = 4$
  5) $S = \{s,u,v,x,y\}$ and $d(s) = 0,\ d(u) = 1,\ d(v) = 2,\ d(x) = 2,\ d(y) = 3$.
     $d'(z) = 4$
  6) $S = \{s,u,v,x,y,z\}$ and $d(s) = 0,\ d(u) = 1,\ d(v) = 2,\ d(x) = 2,\ d(y) = 3,\ d(z) = 4$.

- How do we know Dijkstra's Algorithm give the optimal solution?
- Theorem: For any vertex $u \in S$, the path given by Dijkstra is a shortest path.
  Proof Sketch: By induction on the size of S.
  - Basis case: $|S| = 1, S = \{s\}, d(s) = 0$
  - IH: Assume that this theorem is true when $|S| = k$, i.e., k vertices have been explored and their labels are correct distance from s.
  - We want to show that this theorem is also true when $|S| = k + 1$.
    In this step, we add the vertex v, outside of S.
    Why is P_1 the shortest path from s to v?
  - Maybe P_2 is shorter.
  - Then $d(s,y) < l(P_2) < l(P_1) = d'(v)$
    then Dijkstra would have added y!
    $d'(y) = d(x) + l_{(x,y)} < d'(v)$ Contradiction.



- The *diameter* of a graph is the length of the longest shortest path in the graph.
  $D = \max_{u,v\in G} d(u,v)$, where $d(u,v)$ is the length of the shortest path between u and v.
  - Algorithm to find diameter of the graph: run Dijkstra's algorithm on each vertex as source.
    - For $u \in G$
      1) Run $\text{Dijkstra}(u)$
      2) Record $\max(u)$,
         the maximum label seems in this run of Dijkstra

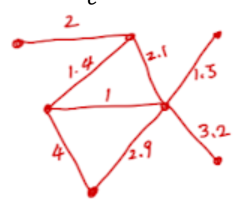
      Endfor
      Return $\max_{u\in G} (\max(u))$

- Implementation
  - If recalculate all labels every time
    Runtime: $O(mn)$, m - # of edges and n - # of vertices
    1) Need to store $d'(v)$s and select the smallest one (to add to S).
    2) After adding a v to S, change the $d'(w)$ for all neighbors w of v.

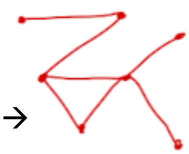
        - Change key d'
        - Extract Smallest – Extract min
    - Priority Que
        1) Change key
        2) Extract min
        - Page 141 – read details on how to use a priority queue to implement Dijkstra's Algorithm.
        - Dijkstra can be implemented using
            - n extract min and m change key
            - $n \log n + m \log n = O(m \log n)$

Minimum Spanning Tree [4.5]

- Input: a weighted (undirected) graph $G = (V, E, w_e > 0)$,
  w_e is the weight on edge e.
- Assume All w_e are distinct.
    - vertices – cities
    - edges – cost of laying a cable between the endpoint cities.

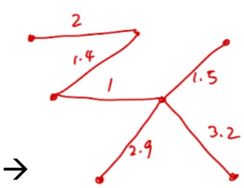


- Can this be the cheapest way to connect all cities? →



    - No, there is cycle.

  The cheapest way to connect all cities. →



- Question: Given $G = (V, E, w_e > 0)$ as input, find the minimum total cost.
  (spanning tree)
    - Spanning tree of a graph is a connected subgraph with no cycle that have all the vertices.

What to expect or prepare for the next class:

- Minimum Spanning Tree

Reading Assignment

Algorithm Design: 4.4, 4.5