Thursday, 19 October 2017

LAB DEMO 07

APIO 2013 TASKSAUTHOR

Cool stuffs about SSSP algorithms

Let's see APIO13_finalversion.pdf, page 7 + 9 (Problem statement 1)

As we have not learned Floyd Warshall's yet,
let's concentrate on **Subtask 5 and Subtask 6 (next week)**

S5: Kill the Optimized Bellman Ford's

```
// pre-condition: the graph is stored in an adjacency list L
counter = 0
for each query p(s,t);
  dist[s] = 0; // s is the source vertex
  loop V-1 times
    change = false;
    for each edge (u,v) in L
      increase counter by 1;
      if dist[u] + weight(u,v) < dist[v]
        dist[v] = dist[u] + weight(u,v);
        change = true;
    if change is false // this is the 'optimized' Bellman Ford
      break from the outermost loop;
  output dist[t];
```

Challenge: Construct a Graph that can still make this Optimized Bellman Ford's run at it's slowest time complexity, i.e. O(VE)

Think First ☺ & Discuss

Wordy Solution
(sample drawing in the next slide)

- The Optimized Bellman Ford's has a check inside the outer for-loop that if there is no more edge relaxation, it will immediately stop.
- In order to force the Optimized Bellman Ford's to really repeat all *E* edges relaxation *V-1* times, we need to observe the listing of the *E* edges.
- The code is written in such a way that all outgoing edges of vertex 0 is processed first, then all outgoing edges of vertex 1, ..., until all outgoing edges of vertex *V-1.*
- Therefore, we need the source vertex to be vertex *V-1*.
- We create a graph where we have edges from vertex *i* to vertex *j* if $j < i$.
- The weight of edge *(i, j)* is 1 if $i-j = 1$ or $i-j+1$ otherwise.
- This way, all *E* edges have to be relaxed exactly $V-1$ times.
- The Modified Dijkstra's has no problem with this input graph. (see next week)

Sample Graph

AND MANY OTHER EDGES FROM i to j if j < i

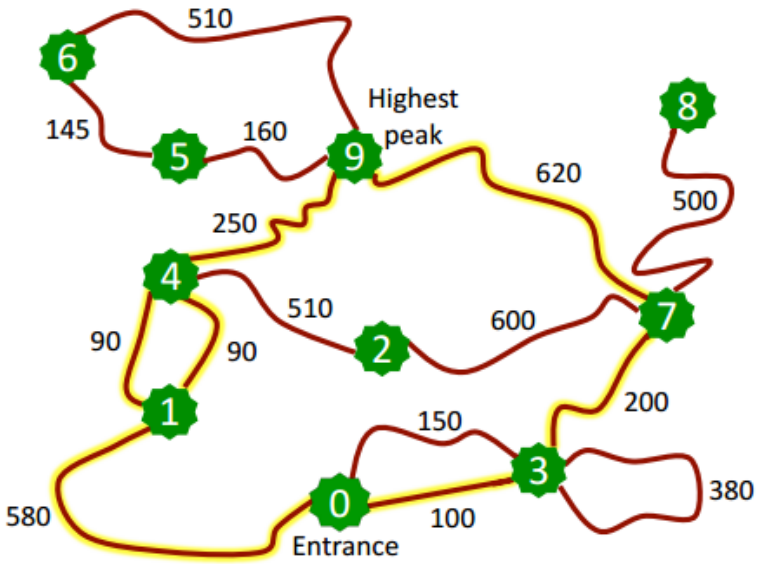
The source is this vertex :O

Try running the Modified Bellman Ford's

Another Graph Modeling…

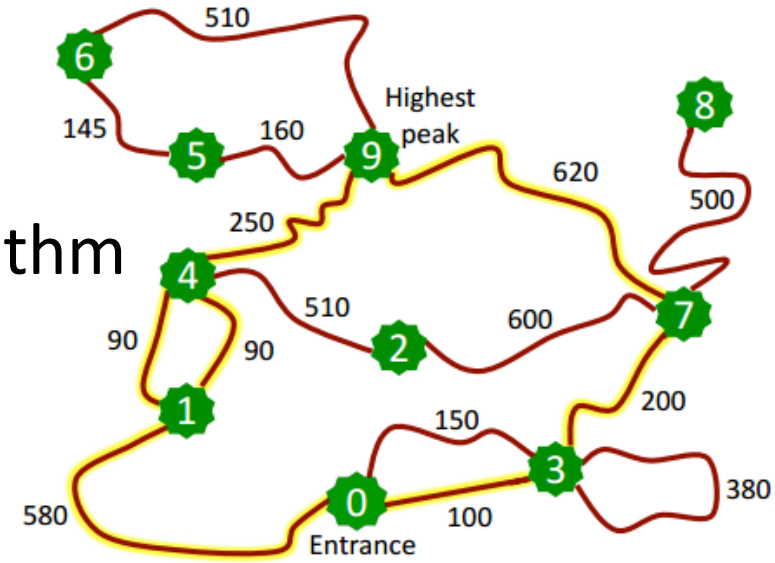
Focus on modeling and solving [UVa 12878] as an SSSP problem

Problem	Verdict	Lang	Time	Best	Rank	Submit Time
12878 - Flowery Trails	discuss	Accepted	Java	1.226	0.000	47	3 mins ago



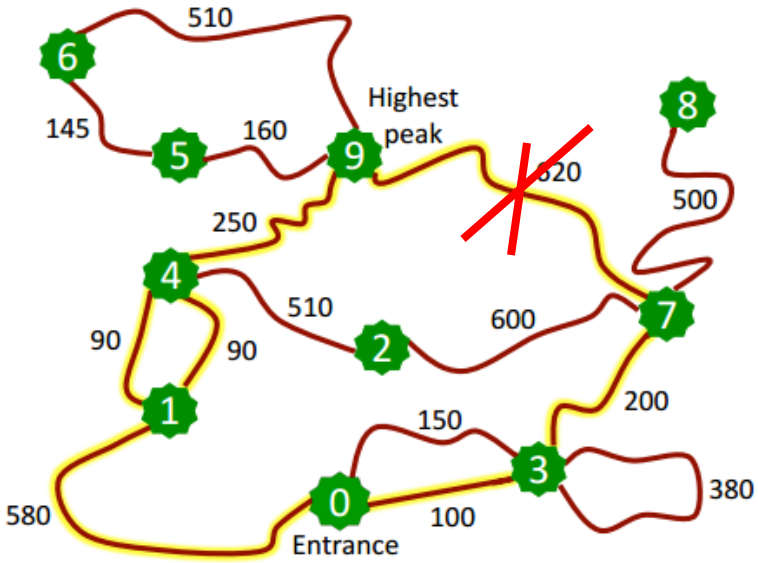
Discussions (1)

- The graph is obvious (it is clearly drawn that way)
- The graph is not simple (e.g. two copies of edge 1-4, just happen to have the same weight)
- The problem is an SSSP problem on weighted graph, no negative weight spotted, e.g. SP from 0 to 9 is 920 units, via 0→3→7→9 or 0→1→4 (2 ways)→9)
- Can we use bellman ford to solve it?
  - **O(VE) = O(PT) = 250000*10000**
  - Too slow !
- Next week will see better algorithm (Djikstra's algo) to solve this



Discussions (2)

- Assuming we solve the SSSP using an efficient algorithm
- Now how to get the length of the yellow edges? (edges that are part of ANY SP from s to t)
- However, any SP spanning tree will only contain edges from 1 possible SP from entrance to highest point, not all edges from all possible SPs, otherwise there will be a cycle (for e.g edge (7,9) is not included in some valid SP spanning tree)



Discussions (3)

- **Observation:** Alternative SPs means that there are alternative edges that can lead to same SP cost.
- **Idea:**
  - For each vertex V, store **all predecessors** that will give the **same current** D[V] cost as the SSSP algorithm is run
  - When better D[V] found, replace with the new predecessor
  - This gives a general SP graph rather than SP spanning tree at the end of the algorithm
  - Then run DFS from highest peak to entrance on the SP graph summing up all edges along the way and *2
- Will revisit this next week after you learn Djikstra's algorithm

Some Shortcuts for OQ2 (2 weeks later)

(most of it provided kindly by your lab TA YuanBin)

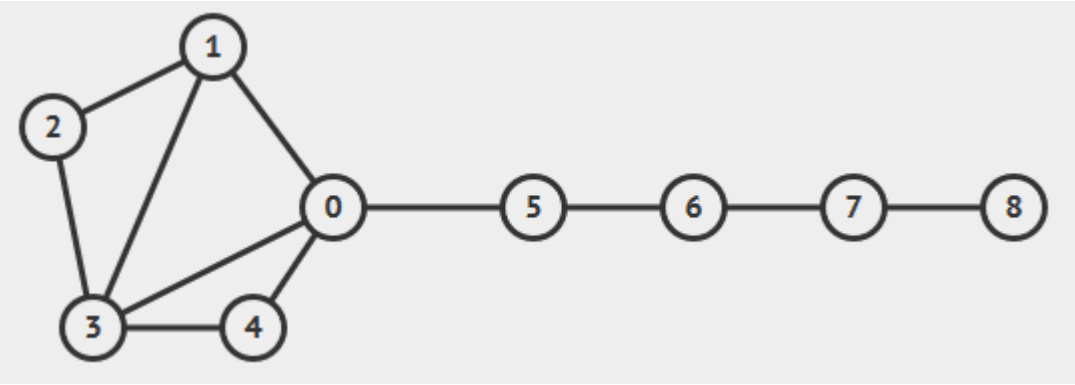
Man versus Machine (1)

Some questions have shortcuts…

- Example 1: How many different spanning trees are there in a complete graph with **V** vertices?
  - Use **Cayley's Formula**… (can't prove this though)
- Example 2: Find the maximum edge weight along the minimax path in a graph
  - Minimize maximum edge weight – Minimum Spanning Tree!
  - Graph traversal along the MST, keep track of the largest edge along the path
  - Similar to your PS4
  - Use Maximum Spanning Tree, and track the minimum for the inverse question (minimum edge weight along the maximin path)

Man versus Machine (2)

- Example 3: Draw a simple connected graph with **V** vertices and **E** edges such that the graph contains **k** cut vertices/bridges
  - Draw a cycle with **V** – **k** vertices
  - Extend the remaining **k** vertices in a straight line
  - Add in edges inside the cycle as necessary
  - Eg for **V** = 9, **E** = 11, **k** = 4



PS4, Last Minute Discussion

Lab TA will stay back to answer last minute question, if you have any

Name	A	B	C	D
?	AC	AC	AC	AC
?	AC	AC	AC	
?	AC	AC		
?	AC			
The rest?				