CS2010 Semester 1 2012/2013

Data Structures and Algorithms II

E-Tutorial 06 - Shortest Paths 1

For Week 08 e-Tutorial (08 October - 12 October 2012)

Released: Wednesday, October 3, 2012

Document is last modified on: October 10, 2012

1 Introduction and Objective

Week08 of this semester is SoC turn to execute the NUS e-Learning week. It is an NUS policy post-SARS that NUS teaching staffs should be prepared to conduct classes online in case of a major outbreak/disaster prevents students from physically coming to NUS.

For lecture, Steven is too tired from his Italy trip to produce a new (better) video, so he will just ask student's to view last year's version (only minor modifications compared to this year's slides).

`http://ivle.nus.edu.sg/bank/media/modvideo.aspx?KEY=`
`7e528a4f-32bd-47a5-8619-d66ea438f0b7&ChannelID=22adcec0-b0d2-4ee1-a102-9611a4d61c45`

For tutorial, it is a bit harder to make this into an e-tutorial, as tutorial rely on heavy interaction between TAs and students. So, read and attempt the questions in this tutorial first. During our designated tutorial time, we (the tutors) will standby in front of our computer to discuss the questions over this IVLE chat room.

`https://ivle.nus.edu.sg/chatroom/chatroom.aspx?`
`ChatID=99b30f19-d622-42af-96a0-313eec71e863`

In case you still have any doubts after this e-tutorial... Don't worry. I have planned the course such that the SSSP problem is covered with two lectures and two tutorials... We can clarify doubts during Week09 (or other sessions). Moreover, I have also asked Lab TA to re-discuss question 3 and 4 during their lab demo on Week08. Somehow lab sessions are excluded from e-learning :).

Note: As usual, use `http://www.comp.nus.edu.sg/~stevenha/visualization/sssp.html` to *verify* the answers of some questions in this tutorial. However during written tests, you have to be able to do this by yourself.

2 Tutorial 06 Questions

Shortest Path application

Q1. What are the costs of the shortest path between A and H, A and F, and A and E? Show how you will use Bellman Ford's algorithm taught in Lecture 07 to solve this question. You can assume that the edges are sorted according to increasing vertex pair labels (e.g. (A,B) before (A,C)).

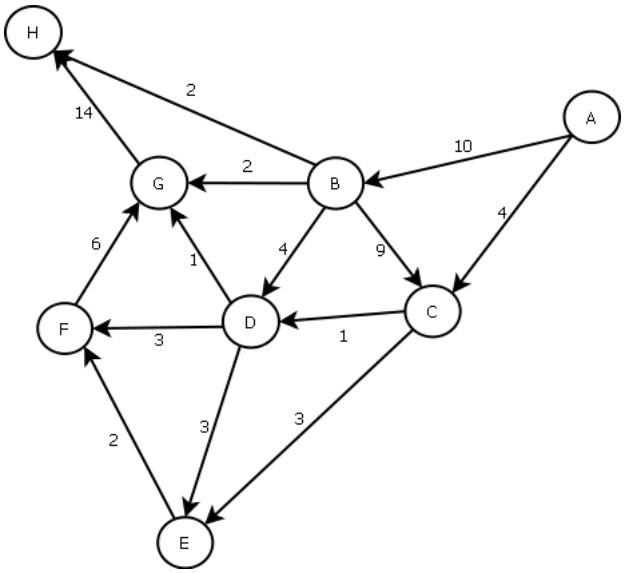

Figure 1:

Ans: We run an SSSP algorithm (Bellman Ford's) from A (the common source between the three pairs). The edges will be processed in this order:
(AB), (AC), (BC), (BD), (BG), (BH), (CD), (CE), (DE), (DF), (DG), (EF), (FG), (GH).

Vtx → Init → 1stpass
A → 0
B → inf → 10 (AB)
C → inf → 4 (AC)
D → inf → 14 (ABD) then 5 (ACD)
E → inf → 7 (ACE)
F → inf → 8 (ACDF)
G → inf → 12 (ABG) then 6 (ACDG)
H → inf → 12 (ABH)

To get the following SP spanning tree (Figure 2).

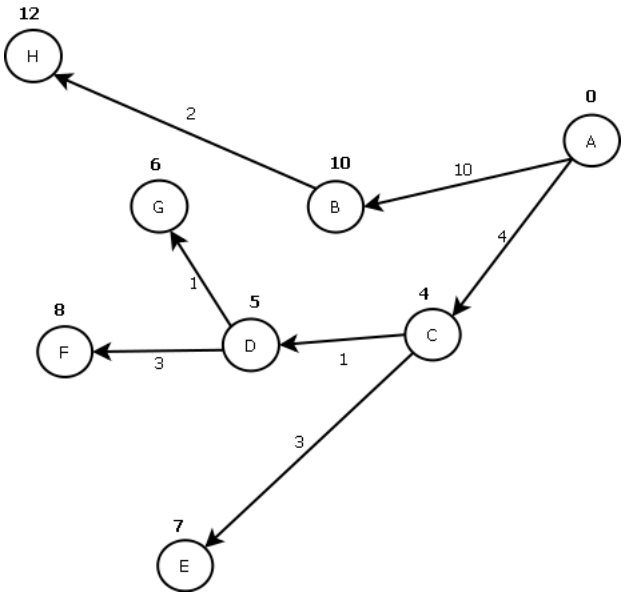

Figure 2:

Then we will get these paths: path $A \rightarrow B \rightarrow H$ with cost 12, path: $A \rightarrow C \rightarrow D \rightarrow F$ with cost 8, and path: $A \rightarrow C \rightarrow E$ with cost 7.

Q2. Using the graph shown in Figure 1, how many times Bellman Ford's passes through all edges before you cannot see any more edge relaxations? Can this phenomenon be used to slightly improve Bellman Ford's algorithm?

After the first pass through all edges, actually the SP spanning tree (Figure 2) is already found. However, Bellman Ford's algorithm should go at least one more round through all the edges to check if there is any more edge relaxation. Since there is no more such edge relaxation, it means that no shorter path can be found from A to any other node.

This can be used as a simple improvement in Bellman Ford's implementation. Once there is no more edge relaxation at the current round (the inner loop that goes through all edges), we can terminate Bellman Ford's outer loop. However, the worst case time complexity stays at $O(VE)$.

Graph Modeling Exercises

Q3. Find the graph, identify the underlying shortest paths problem, and choose the best algorithm that you have learned so far to solve the problem.

UVa 12160 - Unlock the Lock
http://uva.onlinejudge.org/external/121/12160.html

Ans:
This problem can be modeled as an SSSP problem on unweighted graph.
But first... where is the graph?
Vertices: Number between [0000..9999], there are up to $V = 10000$ vertices.
Implicit edges: Link two numbers A and B with a unweighted directed edge $A \rightarrow B$
If $B = (A$+certain button value) % 10000; (do you know why we need modulo 10000?).
Each vertex has at most R outgoing edges, one for each button press, or up to $E = 100000$ edges.
Best algorithm: $O(V + E)$ BFS from source L, report the shortest path value at $D[U]$.
Potential Trap: Be careful to report "Permanently Locked" if U is unreachable from L.
Implicit edges - not supplied as part of the input or represented as objects in memory, but rather determined algorithmically from the input.

Q4. Find the graph, identify the underlying shortest paths problem, and choose the best algorithm that you have learned so far to solve the problem.

UVa 929 - Number Maze
http://uva.onlinejudge.org/external/9/929.html

Ans:
This problem can be modeled as an SSSP problem on weighted graph.
Again... where is the graph?
Vertices: (row, col) cells, there are NM cells/vertices.
Implicit (again) edges: North/East/South/West of each cell, weighted based on the value of *target cell*. There are up to 4NM edges. Note that some cells only have two edges (the 4 corner cells), some other cells only have three edges (the side cells, along the border).
Best algorithm: $O(NM \log(NM))$ Dijkstra's algorithm from source (row 0, col 0), report the shortest path value at D[target (row N-1, col M-1)].
Potential Trap: Be careful with the starting value of D[source (row 0, col 0)] is not 0, but value of cell (0, 0). See the interesting test case below:

```
1
5
10
7 0 0 0 0 0 0 0 0 0
9 9 9 9 9 9 9 9 9 0
0 0 0 0 0 0 0 0 0 0
0 9 9 9 9 9 9 9 9 9
0 0 0 0 0 0 0 0 0 0
```

The answer is 7, the value of cell (0, 0), and then the shortest path is obviously the zig-zag path that uses all the zero cells from (0, 0).