CS161 – Fall 2016 — Homework 5

Jenny Zeng, SID 52082740, zhaohuaz@uci.edu

Homework 5, due 9:00pm on Friday, Nov. 4: R-13.4, R-13.12, C-14.2, C-14.4

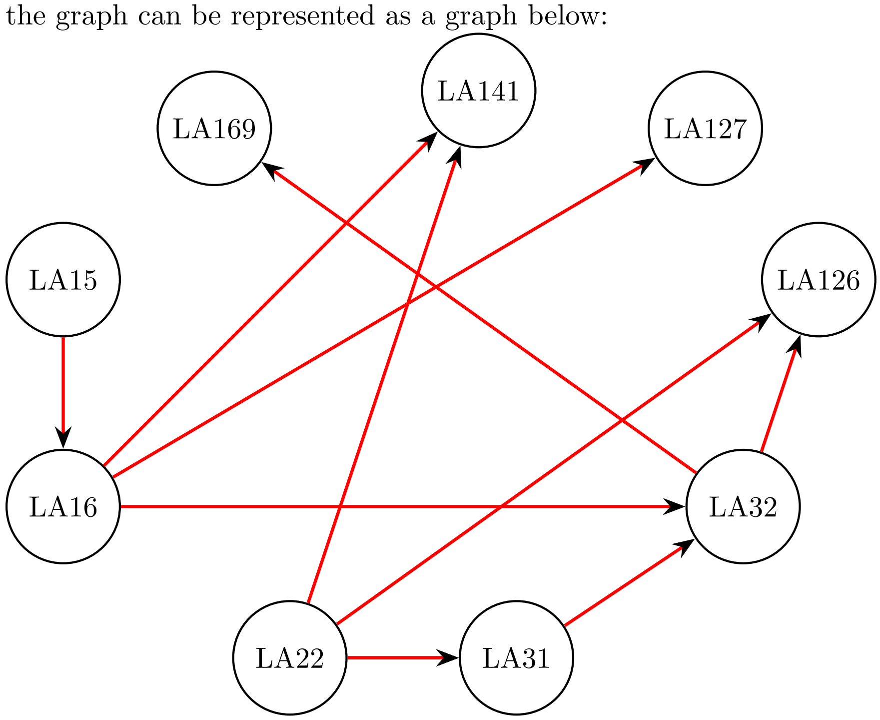
R-13.4

Bob loves foreign languages and wants to plan his course schedule to take the following nine language courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169. The course prerequisites are:

- LA15: (none)
- LA16: LA15
- LA22: (none)
- LA31: LA15
- LA32: LA16, LA31
- LA126: LA22, LA32
- LA127: LA16
- LA141: LA22, LA16
- LA169: LA32.

Find a sequence of courses that allows Bob to satisfy all the prerequisites.

Answer:
one possible sequence is: LA15, LA16,LA22,LA31,LA32,LA169,LA141,LA127,LA126
the graph can be represented as a graph below:



R-13.12

Give the order in which the edges are labeled by the DFS traversal shown in Figure 13.5.

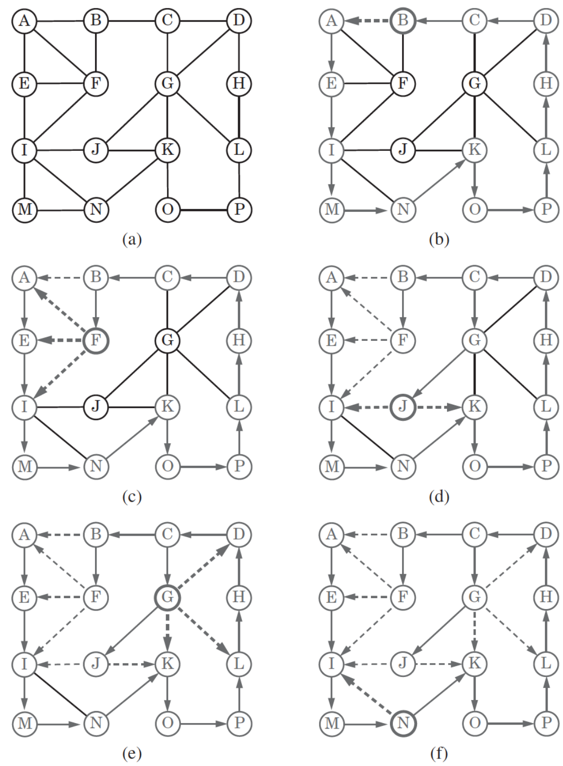


Figure 13.5: Example of depth-first search traversal on a graph starting at vertex A. Discovery edges are drawn with solid lines and back edges are drawn with dashed lines. The current vertex is drawn with a thick line: (a) input graph; (b) path of discovery edges traced from A until back edge (B,A) is hit; (c) reaching F, which is a dead end; (d) after backtracking to C, resuming with edge (C,G), and hitting another dead end, J; (e) after backtracking to G; (f) after backtracking to N.

Answer:
edges labeled in each graph can be represented as below:

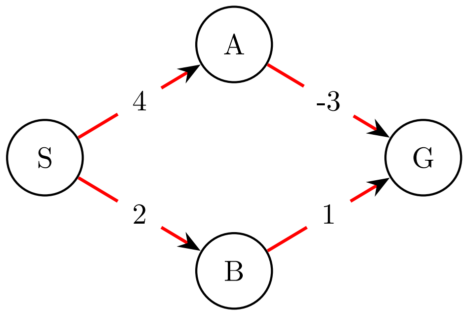
- (b): path of discovery edges traced from A until back edge (B,A) is hit
  (A,E),(E,I),(I,M),(M,N),(N,K),(K,O),(O,P),(P,L),(L,H),(H,D),(D,C),(C,B),(B,A)
- (c): reaching F, which is a dead end;
  (B,F),(F,A),(F,E),(F,I)
- (d): after backtracking to C, resuming with edge (C,G), and hitting another dead end, J
  (C,G),(G,J),(J,I),(J,K)
- (e): after backtracking to G
  (G,D),(G,K),(G,L)
- (f): after backtracking to N.
  (N,I)

C-14.2

Give an example of a weighted directed graph, $\vec{G}$, with negative-weight edges, but no negative-weight cycle, such that Dijkstras algorithm incorrectly computes the shortest-path distances from some start vertex v.

Answer:

Example graph:



1. initialize

vertex	S	A	B	G
Distance	0	∞	∞	∞
Parent				
processed?				

2. remove s from Q. Relax edge (S,A) and (S,B)

vertex	S	A	B	G
Distance	0	4	2	∞
Parent				
processed?	$\surd$			

3. remove B from Q. Relax edge (B,G)

vertex	S	A	B	G
Distance	0	4	2	3
Parent		S	S	B
processed?	$\surd$		$\surd$	

4. remove G from Q. No edges to relax.

vertex	S	A	B	G
Distance	0	4	2	3
Parent		S	S	B
processed?	$\surd$		$\surd$	$\surd$

5. remove B from Q. No edges to relax because G is not in Q.

vertex	S	A	B	G
Distance	0	4	2	3
Parent		S	S	B
processed?	$\surd$	$\surd$	$\surd$	$\surd$

The shortest path found by Dijkstras algorithm is $S \to B \to G$ with distance 3 while the actual shortest path is $S \to A \to G$ with distance 1.

C-14.4

Consider the following greedy strategy for finding a shortest path from vertex start to vertex goal in a given connected graph.

1. Initialize $path$ to $start$.
2. Initialize $VisitedVertices$ to $start$.
3. If $start = goal$, return $path$ and exit. Otherwise, continue.
4. Find the edge $(start, v)$ of minimum weight such that v is adjacent to $start$ and v is not in $VisitedVertices$.
5. Add v to $path$.
6. Add v to $VisitedVertices$.
7. Set $start$ equal to v and go to step 3.

Does this greedy strategy always find a shortest path from *start* to *goal*? Either explain intuitively why it works, or give a counterexample.

Answer:
It does not always find a shortest path from *start* to *goal*. In the example graph below, the greedy strategy will only go through edges (S,B), (B,C), (C,G) and get the path $S \to B \to C \to G$ with distance 8 while the actual shortest path is $S \to A \to G$ with distance 6.