1. You are helping a group of ethnographers analyze some oral history data they have collected by interviewing members of a village to learn about the lives of people who lived there over the last two hundred years. From the interviews, you have learned about a set of people, all now deceased, whom we will denote P1, P2, P3,..., Pn. The ethnographers have collected several facts about the lifespans of these people:

* a)  Pi died before Pj was born
* b)  Pi and Pj were both alive at the same time

However, the ethnographers are not sure that there facts are correct. So they’d like you to determine whether the data they have collected is at least internally consistent, in the sense that there could have existed a set of people for which all the facts simultaneously hold. Describe an algorithm to answer the ethnographers’ problem. Your algorithm should output whether all the facts are consistent, or output the subset of facts that may be inconsistent. Hint: construct a bipartite directed graph in which there are 2n nodes representing, respectively, dates of birth and death of the n people. Draw edges from one node to another when we have a fact that states that the former precedes the latter. Clearly a person’s birth precedes his/her death, so that would be an edge. What does it mean if two people are claimed to have been alive at the same time? -> that means those two lines are overlapping

We need to construct a directed graph G in this problem. The test for consistency will then turn to test whether G is acyclic or not.

For each person , we define that nodes and to represent their birth and death dates. Edges in the graph will be corresponding to one event preceding another. So, from the beginning we need to include edges (, ) for each person i.

Based on the two kinds of facts, we also need to include edges (, ) and (, ) to complete construction of G.

Then we can prove it with two different cases:

Assume that G has a cycle, then each of the events corresponding to nodes in this cycle must precede the text. But this means that there is no event among these that can be pit first time, consistent with the given information. So, the information is not internally consistent.

Assume that G has no cycles, then it has a topological ordering. If we use this as the order of birth and death dates of these people, then we would have an ordering that is consistent with all the given pieces of information.

2. The **clustering coefficient** *C(G)* of a simple graph *G* is the probability that if *u* and *v* are neighbors and *v* and *w* are neighbors, then *u* and *w* are neighbors, where *u*, *v*, and *w* are distinct vertices of *G*. 7a. We say that three vertices *u*, *v*, and *w* of a simple graph *G* form a triangle if there are edges connecting all three pairs of these vertices. Find a formula for *C(G)* in terms of the number of triangles in *G* and the number of paths of length two in the graph. [*Hint:* Count each triangle in the graph once for each order of three vertices that form it.]

7b. Explain what the clustering coefficient measures in each of these graphs. - the Hollywood graph (used for the six degrees of Kevin Bacon problem) **-** the graph of Facebook friends - the graph representing the routers and communications links that make up the worldwide Internet

(a): The clustering coefficient is given as:

C(G) =

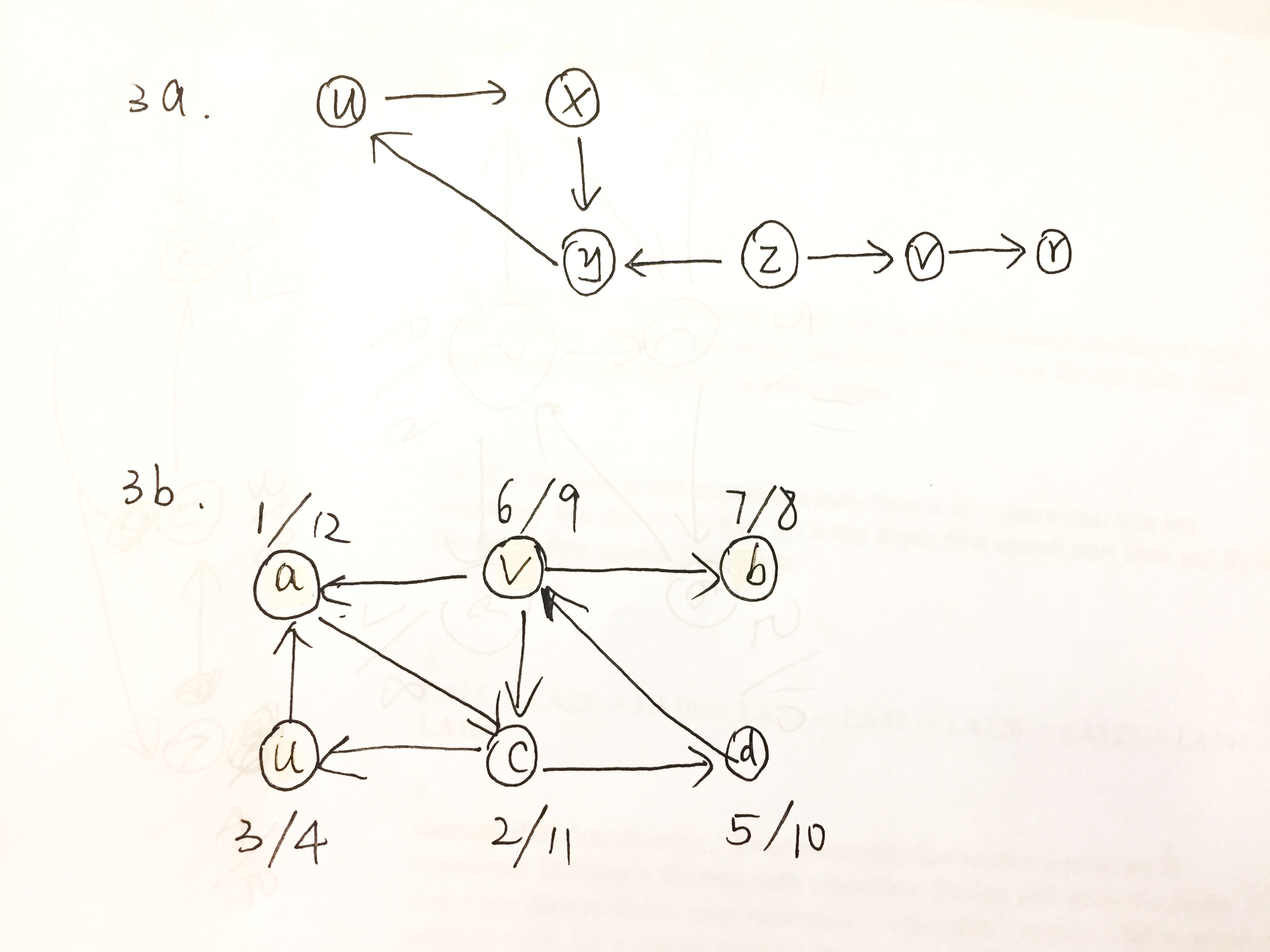
(b): (i): So, it will measure the probability that two actors each of whom has appeared in a film with a random chosen actor have appeared in a film together in the Hollywood graph.

(ii): In the Facebook case, it is used to find the probability that two randomly chosen person’s Facebook friends, it is used to find the probability that two of a randomly chosen person’s Facebook friends are themselves friends.

(iii): In the worldwide internet case, we will measure the probability of two routers each of that has a Communications. So we link to a randomly chosen router and linked among themselves, in the graph representing the routers and communications links that make up the worldwide internet.

3.a. Show how on a directed graph that depth first search starting at vertex u can result in vertex v not being reachable from u even though both u and v have both incoming and outgoing edges.

b. If a directed graph contains a path from u to v, show that it is not necessary that the s(v) < f(u). s() is the depth first search start time and f() is the depth first search finish time



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) LA22: (none) LA32: LA16 and LA31 are prerequisites LA127: LAl6 is prerequisite LA169: LA32 is prerequisite Find a sequence of courses that allows Bob to satisfy all the prerequisites.

LA15 -> LA22 -> LA16 -> LA31 -> LA32 -> LA126 -> LA127 -> LA141 -> LA169

5. Prim’s and Kruskal’s algorithms both “grow” a minimum spanning tree of a graph by selecting edges to add to the tree in a specified, greedy order. Design an efficient algorithm to “prune” a graph and yield a minimum spanning tree by removing edges from the graph in a specified, greedy order. Prove optimal substructure and the Greedy Choice Property.

6.

Shortest Path Verification - Your roommate has written a program to implement Dijkstra’s shortest path algorithm. Design and analyze a linear time algorithm to check your roommate’s algorithm’s results. That is, given a graph G = (V, E), a source vertex s, and your roommate’s values of v.d and v.pi for every vertex v ∈ V , your algorithm must verify their correctness or find a value that is wrong.

Firstly, we need to check that s.d = 0 and s.pi = NIL.

Second, we need to check that v.d = v.pi.d + w(v.pi, v) for all v excepts s.

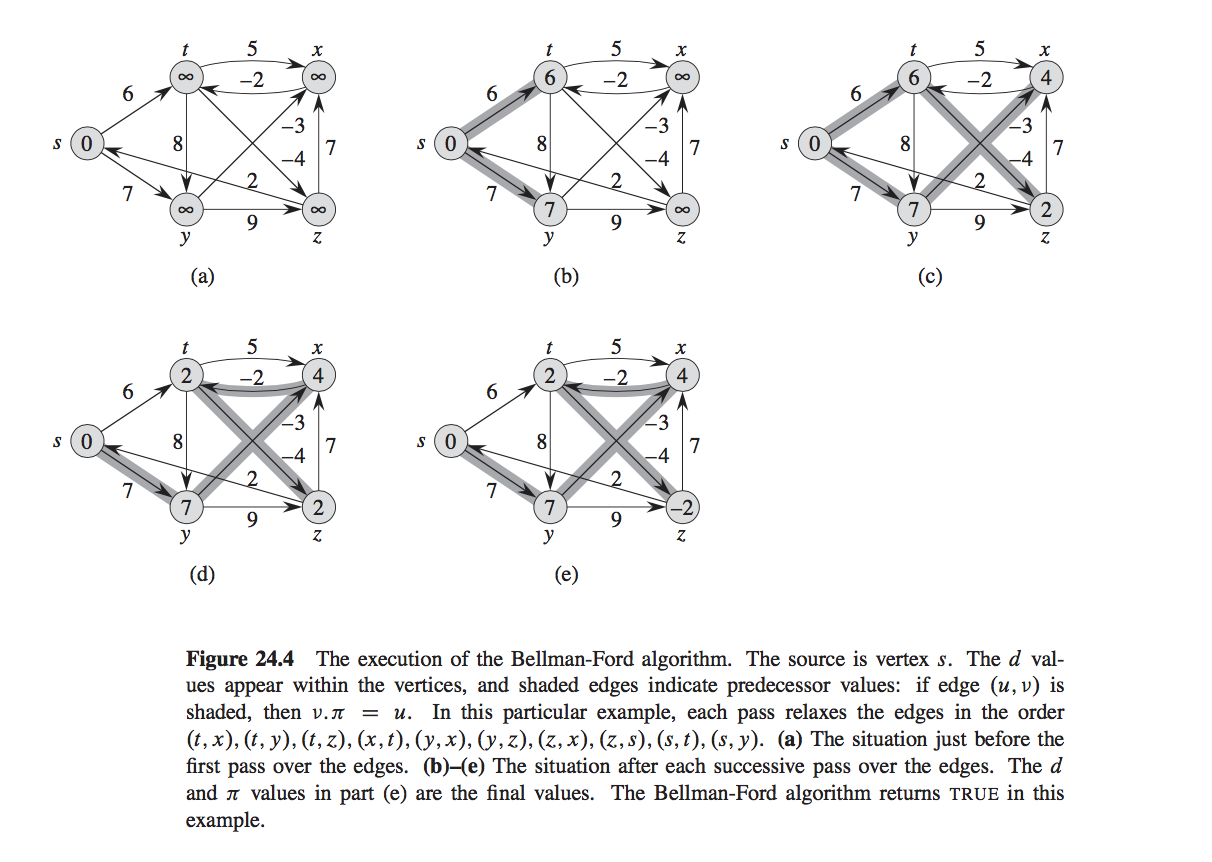
Third, we need to check that v.d = ∞ iff v.pi = NIL for all v excepts s.

If any of the above check tests failed, we can declare that the output for roommate’s algorithm is wrong, otherwise, we need to run one pass of Bellman-Ford. If any values of v.d doesn’t match original values, then we can declare that the output is wrong.

Otherwise, the algorithm should be correct.

7.

Run the Bellman-Ford algorithm on the directed graph of Figure 24.4, using vertex *z* as the source. In each pass, relax edges in the same order as in the figure, and show the *d* and π values after each pass.



d values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | x | y | z |
| ∞ | ∞ | ∞ | ∞ | 0 |
| 2 | ∞ | 7 | ∞ | 0 |
| 2 | 5 | 7 | 9 | 0 |
| 2 | 5 | 6 | 9 | 0 |
| 2 | 4 | 6 | 9 | 0 |

Π values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | x | y | z |
| NIL | NIL | NIL | NIL | NIL |
| z | NIL | z | NIL | NIL |
| z | x | z | s | NIL |
| z | x | y | s | NIL |
| z | x | y | s | NIL |

8.

D(0) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | ∞ | 1 |
| B | ∞ | 0 | ∞ | 5 | 4 |
| C | ∞ | 3 | 0 | 6 | ∞ |
| D | 5 | ∞ | 4 | 0 | ∞ |
| E | ∞ | ∞ | ∞ | 1 | 0 |

D(1) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | ∞ | 1 |
| B | ∞ | 0 | ∞ | 5 | 4 |
| C | ∞ | 3 | 0 | 6 | ∞ |
| D | 5 | 8 | 4 | 0 | 6 |
| E | ∞ | ∞ | ∞ | 1 | 0 |

D(2) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | 8 | 1 |
| B | ∞ | 0 | ∞ | 5 | 4 |
| C | ∞ | 3 | 0 | 6 | 7 |
| D | 5 | 8 | 4 | 0 | 6 |
| E | ∞ | ∞ | ∞ | 1 | 0 |

D(3) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | 8 | 1 |
| B | ∞ | 0 | ∞ | 5 | 4 |
| C | ∞ | 3 | 0 | 6 | 7 |
| D | 5 | 7 | 4 | 0 | 6 |
| E | ∞ | ∞ | ∞ | 1 | 0 |

D(4) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | 8 | 1 |
| B | 10 | 0 | 9 | 5 | 4 |
| C | 11 | 3 | 0 | 6 | 7 |
| D | 5 | 7 | 4 | 0 | 6 |
| E | 6 | 8 | 5 | 1 | 0 |

D(5) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 3 | 2 | 2 | 1 |
| B | 10 | 0 | 9 | 5 | 4 |
| C | 11 | 3 | 0 | 6 | 7 |
| D | 5 | 7 | 4 | 0 | 6 |
| E | 6 | 8 | 5 | 1 | 0 |