1 Question 1: Rule Systems (50 pts)

Josh Lim has been working with the police department for a while. After a few months of doing boring fact-checking work, he decides to get smart and write a computer program that will do his job for him. Through careful introspection he has come up with the following rules to determine whether or not his suspects are likely to be guilty of a crime.

```
P1: (IF ((? x) is a businessman)
        THEN ((? x) makes millions))

P2: (IF ((? x) is in the mob)
        THEN ((? x) has low taxes))

P3: (IF (AND ((? x) has low taxes)
             ((? x) makes millions))
        THEN ((? x) embezzles money))

P4: (IF ((? x) embezzles money)
        THEN ((? x) is a criminal))

P5: (IF (AND ((? x) knows (? y))
             ((? x) makes millions)
             ((? y) is in the mob))
        THEN ((? x) has low taxes)
              ((? x) gets out of jail))
```

One day, a fellow officer, Woody Hoburg, approaches Josh with a few striking facts. Josh tells Woody to give him the list and says he'll get back to him. These are the facts:

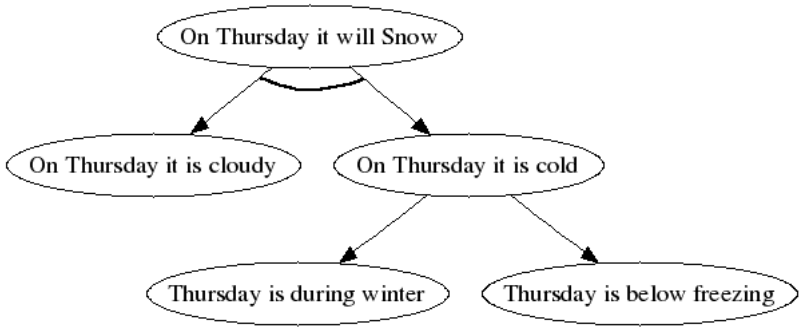
```
A1: (Gotti is in the mob)
A2: (Cagoni is a businessman)
A3: (Cagoni knows Gotti)
```

Goal Trees

An example goal tree is shown below using the following rules:

```
P1: (IF (AND (On (? x) it is cloudy)
              (On (? x) it is cold))
        THEN (On (? x) it will snow))

P2: (IF ((?x ) is during winter)
        THEN (On (? x) it is cold))

P2: (IF ((?x ) is below freezing)
        THEN (On (? x) it is cold))
```

Don't forget that facts are nodes and edges indicate either an AND or an OR relationship. A goal tree for the proposition *(On Thursday it will snow)* will look like:



1.1 Backward Chaining (30 pts)

Essential Assumptions for backward chaining:

1) When working on a hypothesis, the backward chainer tries to find a matching assertion in the database. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. In case none are found, then the backward chainer assumes the hypothesis is false.

2) The backward chainer never alters the database, so it can derive the same result multiple times.

3) Rules are tried in the order they appear.

4) Antecedents are tried in the order they appear.

Josh first wants to determine who is a criminal. Seeing that Gotti is a mobster, he starts his investigation there. Draw the goal tree for the statement *Gotti is a criminal.* Partial credit will be given for partial completion of the goal tree.

Draw the Goal Tree

(Gotti is a criminal)

What additional fact is necessary to conclude that Gotti is a criminal?

.

Simulate backward chaining from the hypothesis *(Cagoni gets out of jail).*

Write all the hypotheses that the backward chainer looks for in the database in the order that the hypotheses are looked for. The table has more lines than you need. **The goal tree will help us assign partial credit** in the event you have made a mistake in the list.

1)
2)
3)
4)
5)
6)

Goal Tree:

(Cagoni gets out of jail)

Does Cagoni get out of jail?

.

1.2 Forward Chaining (20 pts)

Essential assumptions for forward chaining:

- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the dataset
- If a particular rule matches assertions in the database in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule has an antecedent that matches both A1 and A2, the match with A1 is considered first.

Josh augments the original rule set with the following rule:

```
P6: (IF ((? x) gets out of jail)
        THEN ((? x) is not a criminal))
```

Josh then performs forward chaining using the augmented rule set on the original facts. Circle the rules that match the initial database.

P1	P2	P3	P4	P5	P6

What fact is *first added or deleted* from the database in this step?

.

In the second step, circle the rules that **match** at the second iteration of the algorithm.

P1	P2	P3	P4	P5	P6

What new fact is *added or deleted* from the database in the second step?

.

When Josh continues to perform forward chaining, he notes a problem. What is it?

.

QUESTION 2: SEARCH (50 pts)

Part 1 Basic Search (25 pts)

You and some 6.034 TAs are playing a game where the object is to build up a 6-letter word starting with a single letter and adding or subtracting one letter each turn. To make the game more interesting, each intermediate step along the way must also be a word, and all the words are limited to those in a specified list (O, ON, OW, ONE, THROW, TON, TONE, TONES, TONS, TOW, TOWN, TOWNS, TOWS, TROW, TROWS, and THROWS).

As you ponder the game, you realize that this is really just a search problem, as shown in the diagram on the next page. Each node is a word, with edges connecting nodes if you can move from one word to the other in a single turn. To make it easier for you later, the diagram contains progressive levels corresponding to word length. Within each level, the words are shown in lexicographic order (dictionary order). Be sure to remember that in your dictionary, the word 'dog' would come before 'dogs' or 'dogmatic'!

Part 1A (10 points):

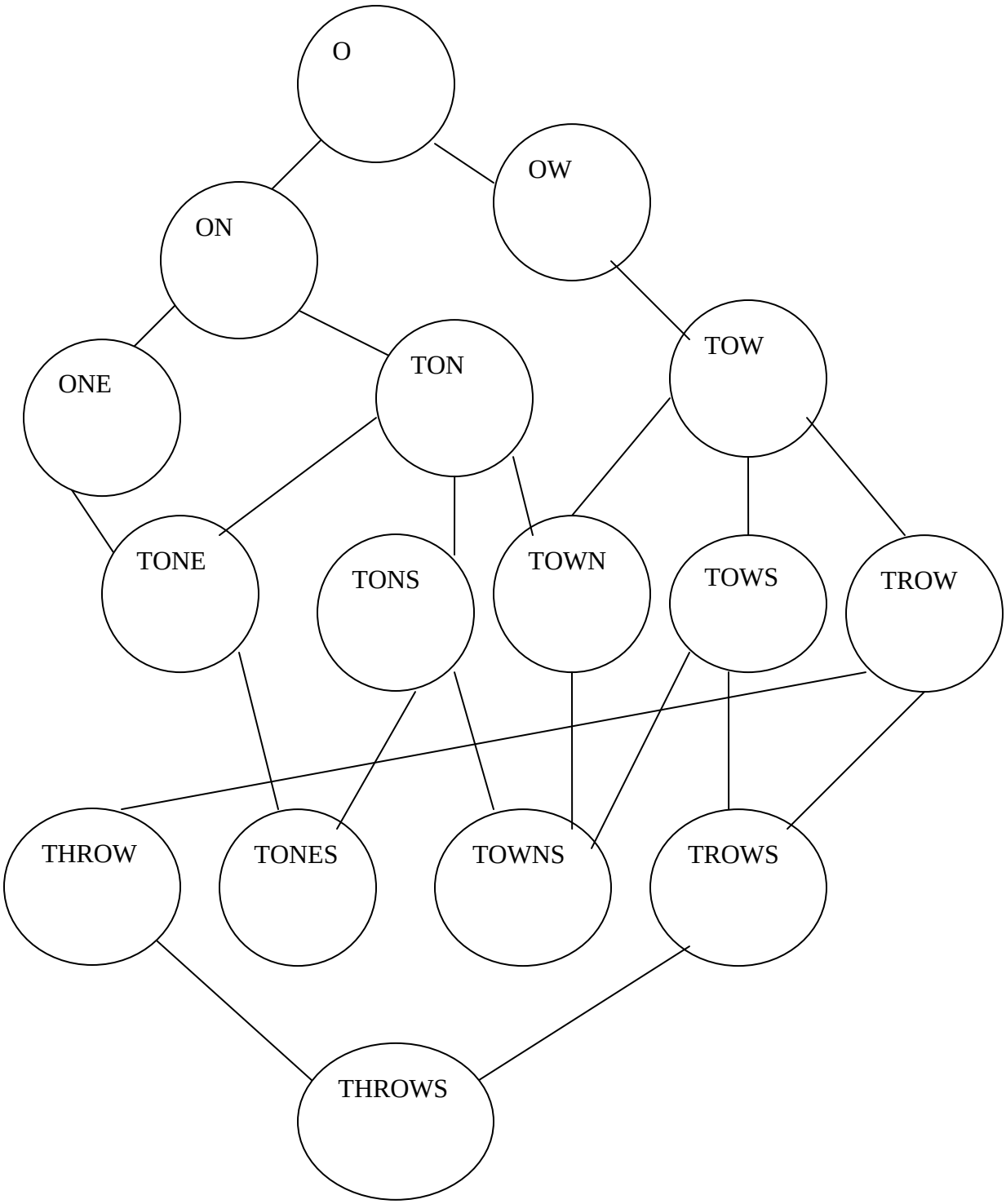
Mike thinks that Breadth-First Search ***with an extended list*** should find us a simple path through the graph. What path does he produce?

Part 1B (15 points):

Mark insists that Depth-First Search ***with backtracking and an extended list*** would be faster. When there is more than one way to extend a path, Mark will break ties using lexicographic order, and of course, he will never extend his path to include a cycle, so if he sees a node already in the path, he will always skip it.

a. Using this technique, how many times will he be forced to backtrack?

b. What will his final path be?


O
OW
ON
TOW
ONE
TON
TONE
TONS
TOWN
TOWS
TROW
THROW
TONES
TOWNS
TROWS
THROWS

Part 2 Optimal Search (25 pts)

One day, for apparently no reason, Geoffrey decides that it would be a great idea to solve the optimal search problem. Jason attempts to convince Geoffrey that they should both already know how to solve optimal search using 6.034 techniques, but Geoffrey doesn't remember it. Geoffrey quickly sketches a graph on the back of a napkin (see Fig 2) and challenges Jason to use his 6.034 know-how to find a solution. Help Jason by using optimal search to find the shortest path from the Start node, S, to the Goal node, G. Break all ties in lexicographic order.

Part 2A (15 points):

Jason decides uses A* search to find a path from S to G, using an extended list. He notes that Geoffrey has helpfully drawn in heuristic values at each node in parentheses, so A* will be easy. Jason asks for your help, and he suggests that you draw a search tree to help visualize the process, and to help provide you partial credit if you make a mistake.

a. The tree

b. What is the final path that Jason will find?

c. How many paths will Jason extend during his search (remember to count the right thing—when he extends the path 'S' and retrieves the three paths 'SA', 'SB', 'SC', that is still only one path extended!)

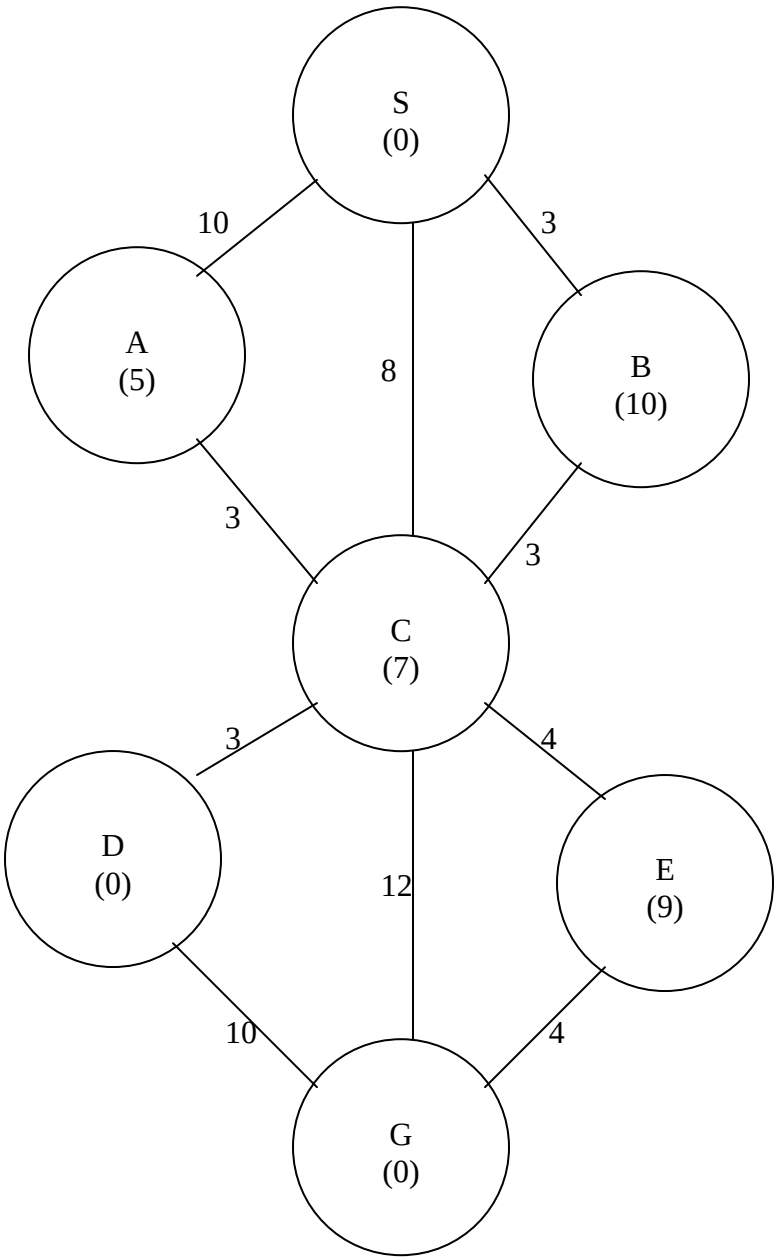


Fig 2

Part 2B (10):

When Geoffrey sees the answer you and Jason give him, he scoffs and crumples it up, throwing it in the trash. Confused, Jason speed-dials Victoria and tells her about his troubles with A* search.

Victoria explains to him what the problem was and suggests using a simpler branch-and-bound search, extending always the shortest path, with an extended list, again using lexicographic ordering to break ties.

a. What path does Jason find this time?

b. What was Victoria's explanation? Be specific.

Tear off sheet, for your convenience, you need not hand this in

The rules:

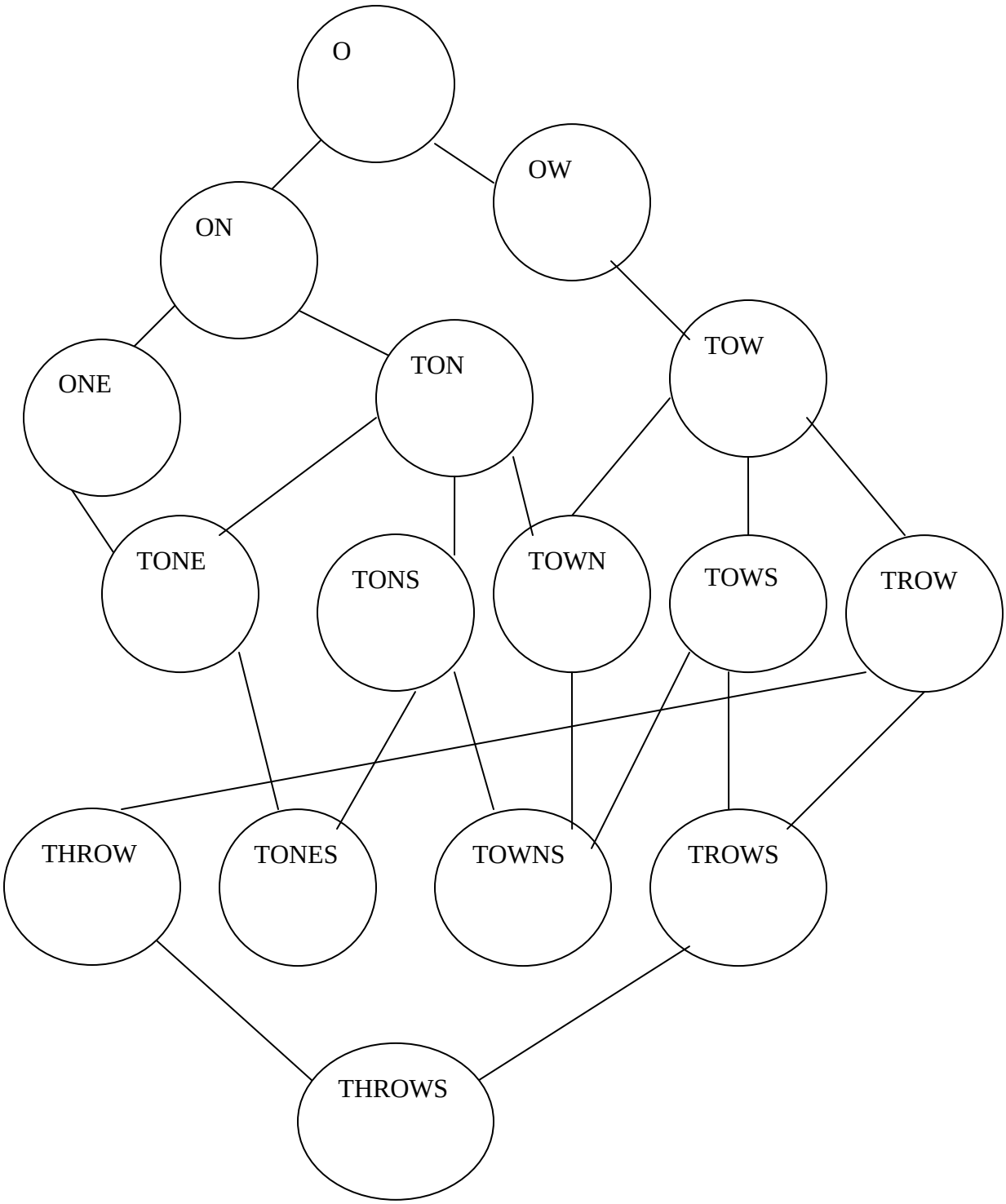
```
P1: (IF ((? x) is a businessman)
        THEN ((? x) makes millions))

P2: (IF ((? x) is in the mob)
        THEN ((? x) has low taxes))

P3: (IF (AND ((? x) has low taxes)
             ((? x) makes millions))
        THEN ((? x) embezzles money))

P4: (IF ((? x) embezzles money)
        THEN ((? x) is a criminal))

P5: (IF (AND ((? x) knows (? y))
             ((? x) makes millions)
             ((? y) is in the mob))
        THEN ((? x) has low taxes)
              ((? x) gets out of jail))
```

The facts:

```
A1: (Gotti is in the mob)
A2: (Cagoni is a businessman)
A3: (Cagoni knows Gotti)
```

Rule added in part 2:

```
    P6: (IF ((? x) gets out of jail)
            THEN ((? x) is not a criminal))
```


O
OW
ON
TOW
ONE
TON
TONE
TONS
TOWN
TOWS
TROW
THROW
TONES
TOWNS
TROWS
THROWS

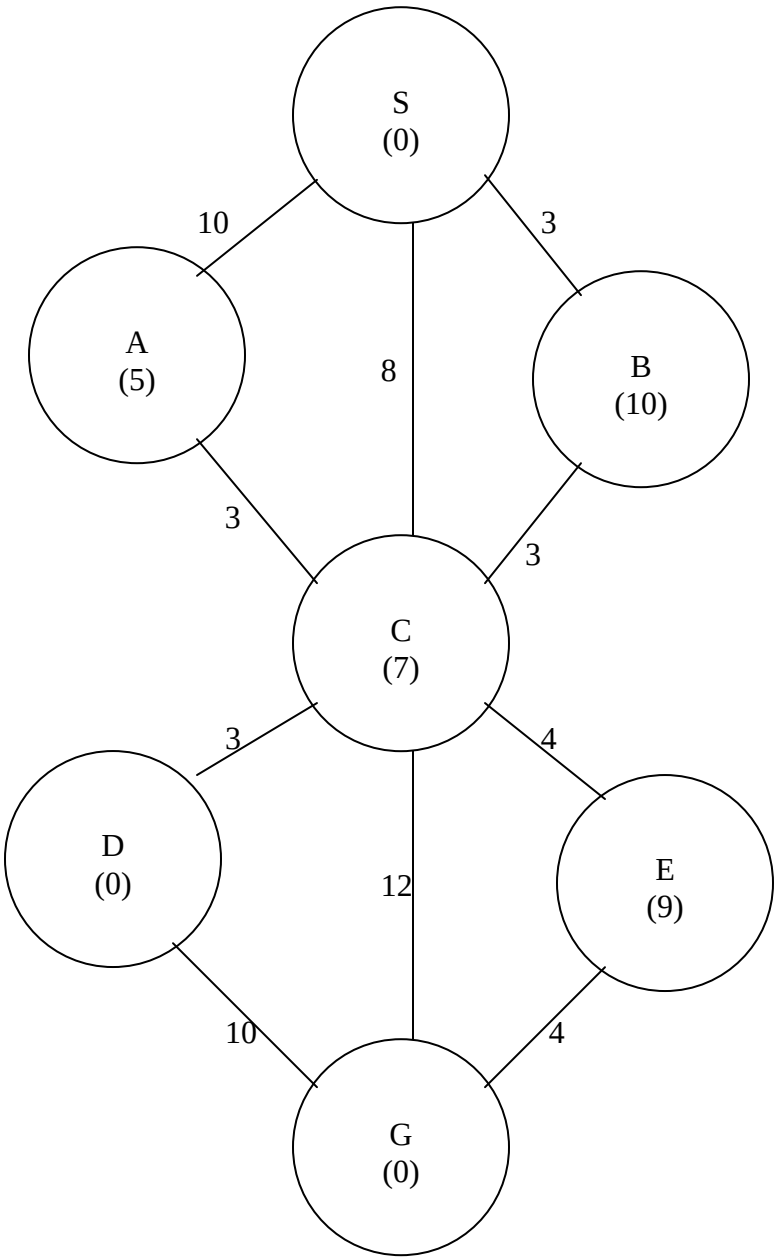


Fig 2