

# CS 367 Homework 6

Alex Pizzuto

---

## 1 Question One

We are given the following incomplete definition of the `Graphnode` class:

```
class Graphnode<T> {  
  
    private boolean visitMark;  
    private List<Graphnode<T>> successors;  
  
    public boolean getVisitMark() {  
        return visitMark;  
    }  
  
    public void setVisitMark(boolean mark) {  
        visitMark = mark;  
    }  
  
    public List<Graphnode<T>> getSuccessors() {  
        return successors;  
    }  
}
```

And we write the `hasSelfCycle` method:

```
public boolean hasSelfCycle( Graphnode<T> node ) {  
    return hasSelfCycle(node, node);  
}  
  
public boolean hasSelfCycle(Graphnode<T> node, Graphnode<T> dest) {  
    node.setVisitMark(true);  
  
    for(Graphnode<t> i : node.getSuccessors()) {  
        if (i == dest) {  
            return true;  
        } else if (!i.getVisitMark()) {  
            if(hasSelfCycle(i, dest) {  
                return true; //Check to see if successor leads back to dest  
            }  
        }  
    }  
    return false; //If we make it here, return false  
}
```

## 2 Question Two

We trace Dijkstra's algorithm on the provided graph.

| visited nodes and their<br>shortest distances from start | dist values for nodes in U<br>(only finite values, listed in increasing order) |
|--|--|
| -  | (0,S)  |
| S: 0   | (4,G), (11,H), (33,P)  |
| S: 0, G: 4   | (10,R), (11,H), (11,P)   |
| S: 0, G: 4, R: 10  | (11,H), (11,P), (30,A)   |
| S: 0, G: 4, R: 10, H: 11                                 | (11,P), (30,A)   |
| S: 0, G: 4, R: 10, H: 11, P: 11                          | (13,A)   |
| S: 0, G: 4, R: 10, H: 11, P: 11, A: 13                   | empty  |

## 3 Question Three

Our three topological orderings are:

- A E B F C D
- C B D F E A
- C A D B E F