CS6150: Advanced Programming
Week 08

Instructor: John Augustine, CSE, IIT Madras

Sept 30, 2024

Name & Roll Number:

Welcome to the second half of the course. You will notice a few changes. From now onward, there will be a score for code quality. This includes (a) clarity and good programming practices, (b) including sufficient comments, and (c) checking for input correctness.

1. There are n children in a class numbered 0 to $n-1$. For this assignment, you can assume that $1 \leq n \leq 100$. For any pair of children i and j, $i \neq j$, they form a friendship with probability $p \in [0, 1]$. The parameters n and p are given to you as input in the following manner as the first line of the input.

```
graph 10 0.4
print
done
```

Your task is to generate such an *undirected* graph based on the first line (in adjacency list format). All subsequent lines in the input are commands until the line contains just the single string done. For this first question, the only command that needs to be implemented is the `print` command that prints the graph in adjacency list format (along with some extra statistics). Here is a sample output.

```
Node 0 : 2 4 5 6 7 9
Node 1 : 2 6
Node 2 : 0 1 5
Node 3 : 5 7 8 9
Node 4 : 0 6 7 8 9
Node 5 : 0 2 3
Node 6 : 0 1 4 7 9
Node 7 : 0 3 4 6 9
Node 8 : 3 4 9
Node 9 : 0 3 4 6 7 8
Avg degree is 4.2
Min degree is 2
Max degree is 6
```

Notice: Since the graph is undirected, each edge (i, j) must occur in two places (j must be in i's list and vice versa). Add comments in the code to indicate where you are taking care of this requirement.

2. The command for this question is `gossip <child number>`, here is an example,

   ```
   graph 10 0.5
   gossip 0
   ```

   Child number 0 (in this case) has learned a juicy bit of gossip (some secret) that no one else knows about. Each day (starting from day 1), all the children who know the secret pass on the secret to a random other node. Note that the other node that a child chooses may already know the secret.

   You must be careful here. If a child learned the secret on day i, she can only pass on that secret starting from day $i + 1$ onward; a careless implementation may allow her to pass it on on day i itself. Add comments in the code to indicate where and how you are taking care of this requirement.

   You must show the progression of the gossip in the following manner by depicting the list of children who learned the information by the end of each day until all children learned the information.

   ```
   day 1: 0 4
   day 2: 0 3 4 7
   day 3: 0 2 3 4 7 8
   ...
   ...
   day 7: 0 1 2 3 4 5 6 7 8 9
   ```

   from its own adj.

3. This is an open ended question that will require some creativity. The command for this question is `flp`. The task now is to find a long path. It is NOT necessary to find the longest path. Your task is to come up with a heuristic to find as long a path as you can. There is one catch. The heuristic that you implement must only be $O(n^2)$ time. It can be randomized, however. The output is the path. For example, for the input

   ```
   graph 5 0.5
   print
   flp
   flp
   ```

   the output may be something like the following.

   ```
   Node 0: 1 3 4
   Node 1: 0 2
   Node 2: 1 3 4
   Node 3: 0
   Node 4: 0 2

   Long path: 0 3
   Long path: 4 2 1 0 3
   ```

   Notice that the first command `flp` produced a path that is not too long, but the second one did. This is acceptable, but of course you should strive to maximize path length.

   Your code must include a paragraph describing your approach in English. Your code must include some comments that indicate that the running time is $O(n^2)$. The comments should include some accounting of how much time each loop is iterating, why your approach is correct and why it is likely to produce long paths, etc.