Submissions: This assignment is due on the 10th of October, 2017. Please note:

1. Each student must submit his or her own assignment.
2. Solutions should preferably be typed in Latex, MSWord or other such word processing software, or printed clearly. In either case, submit a hard copy of your solution. For the programming problem you must write code in either C, C++, Java or Python. Moreover, code should be put in a folder of its own with a README file on how to run the code and test it. It must strictly adhere to the Input/Output format described in the problem below.
3. You must write your name and UUID clearly on your submitted assignment.
4. Staple the pages together. Write your name and UUID on top of each page.
5. It is preferable that you submit solutions to me in class, but it is ok if you can’t do that - just make sure to submit solutions by end of day (i.e., 11:59 PM) by sliding it under my office door (DH 307). For the programming assignment, upload the zipped folder using ecourseware, by the due date.

Academic Integrity: You are encouraged to work in groups, but everyone must write out their own solutions. Absolutely no word to word copying is allowed. If you have worked with other students on the assignment or referred to external sources, please mention all names and sources on your assignment.

Partial solutions: Document your efforts at solving a problem even if you cannot solve it. Write why your approach failed.

DPV ≡ Dasgupta-Papadimitriou-Vazirani Book. JE ≡ Jeff Erickson’s notes

Problem 1[20 pts]: Problem 4.3 from DPV.

Problem 2[80 pts]: Write a program that takes as input a directed graph (in the format described below) and outputs (I) ‘YES’ or ‘NO’ depending on whether the graph is a DAG, and, (II) in case it is a DAG, it outputs a linear ordering of the DAG, and, (III) in case it is a DAG, outputs the length of the longest path in the DAG starting from vertex 1.

The input will consist of several lines and will be given either as a text file or on the command prompt. The first line will be a positive integer \( n \). This is the number of vertices of the graph - the names of the vertices will be 1, 2, 3, . . . , \( n \). The next few lines will contain the edges of the graph as \( i, j \) where \( 1 \leq i \leq n \) and \( 1 \leq j \leq n \) and \( i \neq j \). You can rest assured that the input will be in the correct format and as expected. A valid input could be for example:

```plaintext
5
1, 2
3, 4
3, 1
```

For the input above, one possible correct output is:

```
YES
3, 1, 4, 2, 5
1
```

An example of an incorrect input that you will never see is:

```plaintext
5
0, 1
a, 1
```
If you do not know how to solve the above, read Chapter 4 of DPV.