CSCI 570 Spring 2015 Discussion 2

1. Suppose we have an array A of n integers and we wish to compute an n x n array B where B[i,j] holds the sum of A[i] through A[j]. The following code fragment is one way to solve this problem:

```
for i = 1 to n                  O(n)
    for j = i to n              O(n)          O(n^3)
        sum = 0;
        for k = i to j          O(n)
            sum += A[k];
        B[i, j] = sum;
```

a. What is the running time of the code fragment? Hint: it is $\Omega(n^2)$ but not $O(n^2)$.
b. Do you think this *problem* can be solved in time better than $O(n^2)$? Why or why not?
c. Give an $O(n^2)$ time solution to this problem.

2. Mathematicians often keep track of a statistic called their Erdős Number, after the great 20th century mathematician. Paul Erdős himself has a number of zero. Anyone who wrote a mathematical paper with him has a number of one, anyone who wrote a paper with someone who wrote a paper with him has a number of two, and so forth and so on. Supposing that we have a database of all mathematical papers ever written along with their authors:

a. Explain how to represent this data as a graph.
b. Explain how we would compute the Erdős number for a particular researcher.
c. Explain how we would determine all researchers with Erdős number at most two.

3. In class, we discussed finding the shortest path between two vertices in a graph. Suppose instead we are interested in finding the *longest* simple path in a directed acyclic graph. In particular, I am interested in finding a path (if there is one) that visits all vertices. Given a DAG, give a linear-time algorithm to determine if there is a simple path that visits all vertices.

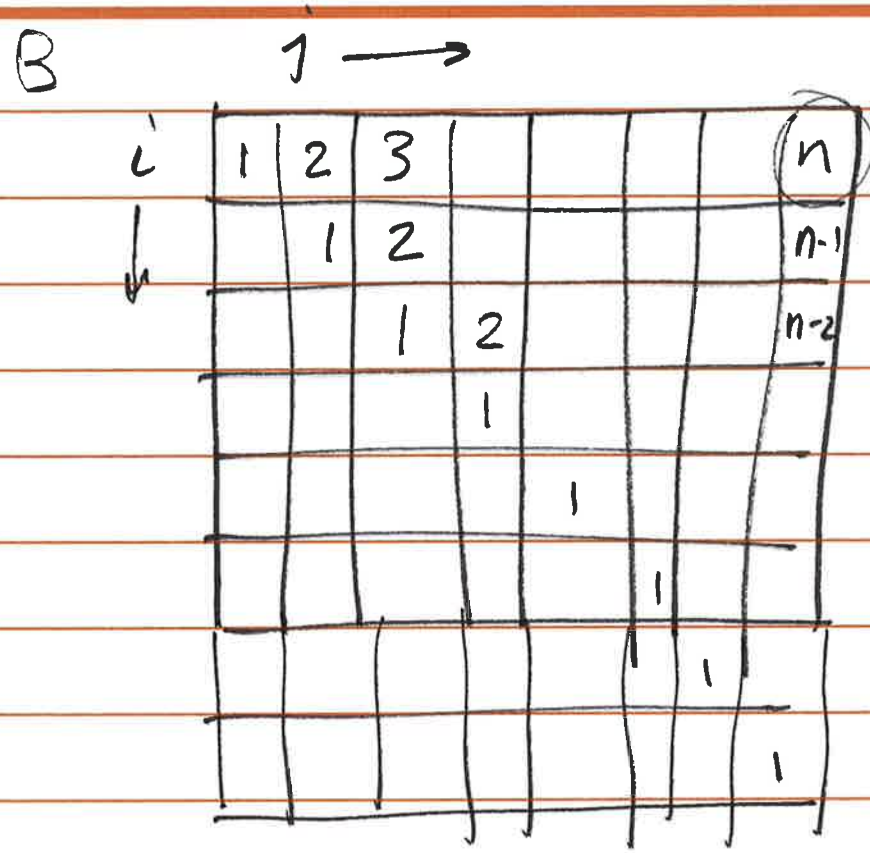
Solution Outlines:

1. a. This is $O(n^3)$.
   b. No; there are $O(n^2)$ output items.
   c. As follows:

```
for i = 1 to n
    B[i,i] = A[i]
    for j = i+1 to n
        B[i,j] = B[i,j-1] + A[j]
```

2. a. Set up a graph with vertices as researchers and (undirected) edges between co-authors.
   b. BFS from Erdos or from the researcher; the layer the other is in is the number.
   c. BFS from Erdos, stopping after layer two.

3. Do a topological sort. If that's a path, it's what we're looking for. If it isn't, no such path exists: it would need to have an inversion relative to that path, and by definition of topological sort, nothing that appears later in the output can have a path to anything former, so *that* wouldn't be a full path either.
   (I'm debating whether or not to explicitly call this a Hamiltonian Path problem).



$$1 \cdot n + 2 \cdot (n-1) + 3 \cdot (n-2) + \dots \; n \cdot 1$$

$$n/2 \cdot n/2 + (n/2-1)(n/2+1) + (n/2-2)(n/2+2) \dots$$

$$n^2/4 + n^2/4 - 1 + n^2/4 - 4 + n^2/4 - 9 \dots$$

$$n^2/4 \cdot n/2$$

$$n^3/8$$

$1 + 4 + 9 + \ldots \left(\frac{n}{2}\right)^2$

$\sum = n^3/24$

$n^3/8 - n^3/24 = \underline{c\,n^3} = O(n^3)$

b- No!

c- for i=1 to n
    B[i,i] = A[i]
    for j = i+1 to n
        B[i,j] = B[i,j-1] + A[j]

- Nodes are researchers
- edges represent co-authorship