Homework 15

Due 11/02/16

October 28, 2016

Develop an *upper bound* for the complexity of the algorithm below, assuming that G is represented using an adjacency list and H is represented using an adjacency matrix. Justify your bound.

```
Input: G = (V, E): graph to analyze
Input: n, m: order and size of G
Output: H: graph with n vertices where the neighbors of each vertex
        are those of distance one or two in G
Algorithm: ExpandedNeighborhood
H = Graph(n)
for v ∈ V do
    for u ∈ N_G(v) do
        H.AddEdge(v, u)
        for w ∈ N_G(u) do
            H.AddEdge(v, w)
        end
    end
end
return H
```

Constructing H takes $\Theta(n^2)$ time, and adding edges are constant time, so your answer to this question depends on how you evaluated the number of times the two loops iterate.

The loosest reasonable bound would be $O(n^3)$, which you can get by noting that a given vertex can be adjacent to at most $n-1$ other vertices in the graph. A somewhat better bound would be if you assumed that the inner loop iterated $\Delta(G)$ (i.e., max degree of G) times, which would give you $O(n^2 + m\Delta)$ complexity, as you can apply the Handshaking Lemma. (This is a better bound than $O(n^2 + n\Delta^2)$.)

The most accurate bound is $\Theta(n^2 + \sum_{v \in V} \deg(v)^2)$. You get this bound by noticing that, for each vertex v, GetNeighbors is called exactly $\deg(v) + 1$ times—once in the outer loop, and one more time when the outer loop hits each of the neighbors of v. There's not a clean, closed-form expression for this summation, though the overall complexity will be n^3 for dense graphs and n^2 for sparse graphs (due to the construction of H).