22C : 031 (CS : 3310 : 0001) Algorithms
Homework 4

This homework is based on our discussions of dynamic programming from Chapter 6 of the text and from our notes.

- Exercise 2 of Chapter 6. (15 points)
- Exercise 6 of Chapter 6. (15 points)
- Exercise 9 of Chapter 6. (15 points)
- Let us define a *layered* graph as a directed graph $G = (V, E)$ in which the vertex set V can be partitioned into disjoint subsets $V_0.V_1, \ldots, V_t$, with the property that *any* edge is from some vertex in V_i to some vertex in V_{i+1} for some i between 0 and $t-1$. We will refer to the V_i as the layers of the graph. Furthermore, the layer V_0 has only one vertex, which we denote by s.

  Describe an algorithm that, given such a graph, computes, for each vertex v in the graph, the *number of paths* from s to v. The algorithm should run in time polynomial in the number of vertices plus edges. You can assume that the graph is given to the algorithm using some convenient representation – for example, an adjacency list representation, in which each vertex has a list of incoming edges, a list of outgoing edges, and knows the layer to which it belongs. (15 points)

  In the example shown on the next page, there are 4 paths from s to the only vertex in V_3; there are 4 paths from s to a, and 8 paths from s to b.

  Hint: Consider a vertex $v \in V_i$. Relate the number of paths from s to v to the number of paths from s to the vertices in V_{i-1} from which there are edges to v.

The homework is due Tuesday, March 29, in class; if you can't make it to class on that day, just make sure you get it to me by that time.

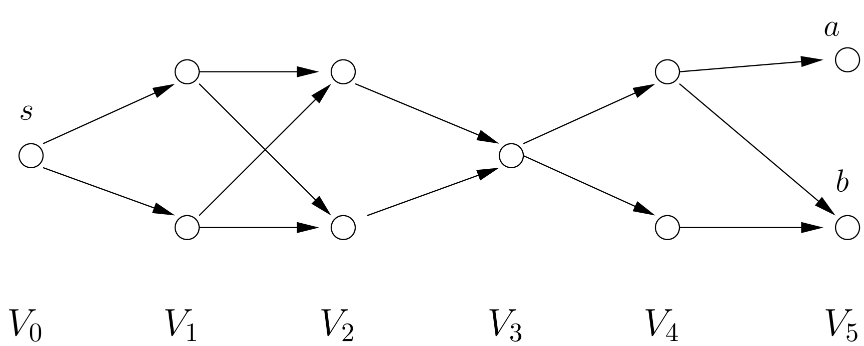

Figure 1: A layered graph example