CS 3510 Fall 2005

Programming Assignment # 2

Due **Tuesday, Nov. 22nd, 11:55pm**

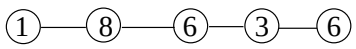
Let $G = (V, E)$ be an undirected graph with n vertices. A subset $S \subseteq V$ of vertices is called an *independent set* if no two vertices in S are joined by an edge in G. The problem of finding large independent sets is difficult in general but it can be done efficiently if the graph is "simple" enough.

A graph $G = (V, E)$ is called a *path* if its vertices form a sequence $v_1, v_2, \ldots, v_k$ and $(v_{i-1}, v_i) \in E$ for $i = 2, 3, \ldots, k$. We associate a positive integer weight w_i with every vertex v_i.

The goal of this assignment is to

Find an independent set in a path G whose total weight is as large as possible.

Below is an example of a path on 5 vertices with weights associated to them.



The maximum weight of an independent set is 14.

Part I: Solve parts (a), (b) and (c) given below, and include the solutions in your report.

a) (15 pts.) Give an example to show that the following algorithm *does* not always find an independent set of maximum total weight.

```
The ''heaviest-first'' greedy algorithm
S := ∅
While V ≠ ∅
pick a node v_i of maximum weight
add v_i to S
delete v_i and its neighbors from G
return S
```

b) (15 pts.) Give an example to show that the following algorithm *does* not always find an independent set of maximum total weight.

```
Let S_1 be the set of all v_i where i is an odd number.
Let S_2 be the set of all v_i where i is an even number.
(Note that S_1 and S_2 are both independent sets.)
Determine which of S_1 or S_2 has greater total weight, and return this one.
```

c) (30 pts.) Describe a dynamic programming algorithm that takes a path G on n vertices with positive integer weights and returns an independent set of maximum total weight. The running time of your algorithm should be polynomial in n, independent of the values of the weights.

Part II: (40 pts.) Implement your efficient algorithm for part (c) as well as a "brute force " algorithm (which can essentially try all possible independent sets to find one of the maximum total weight). For each of the following values of n :

n	5	15	25	35	40

design 10 test cases (i.e. choose the values of weights $\{w_i\}$). Run your two programs using these test cases as inputs and record the running time of the programs. Choose the values w_i so that they are positive integers in the range from 1 to 100. For each of the five values of n, among the 10 test cases, report the average running time, and the maximum running time required by the brute force algorithm and your dynamic programming algorithm, in tabular form.

If a program takes more than 10 minutes you can report it as "undetermined" and terminate it. On the other hand, if a program is able to return an answer immediately, you can report that it takes "zero seconds."

You may earn **15 points** for the implementation of a brute force algorithm, **15 points** for the implementation of your dynamic programming algorithm and **10 points** for the quality of your report.

You may work **in pairs**. You will submit your assignment, **both a report and code via WebCT.** You can earn **100 points** in total.

Submission guidelines: will be posted on class website.