Due at the beginning of class on May 12, 2005. As always, clearly-written partial solutions are welcome, if you cannot solve a problem completely.

Consider Problem 1 in Section 6.12, pages 293-294 of the textbook.

1. Solve parts (a), (b), and (c) of this problem.

2. Implement your efficient algorithm for part (c), as well as a “brute-force” algorithm which does not do anything clever (it can essentially try all possible independent sets etc., to find one of maximum total weight). You are welcome to use any programming language you are comfortable with. For each of the following five values of $n$ (the number of nodes in the path $G$): (i) $n = 5$, (ii) $n = 15$, (iii) $n = 25$, (iv) $n = 35$ and (v) $n = 40$, you have to design 10 test cases (i.e., choose the $\{w_i\}$ values), run your two programs using these test cases as input and record the running time of the programs. Please choose the $w_i$ values so that they are positive integers in the range 1 to 100; otherwise, you have no constraints in choosing them.

Submission requirements: You are required to turn in two things in hardcopy: (a) The printouts of your programs. (b) For each of the five values of $n$, among the 10 test-cases, report the average running time and maximum running time required by the brute-force search algorithm and your dynamic programming algorithm, in tabular form. If a program takes more than 10 minutes to return an answer, you can report it as “undetermined” and terminate that run. On the other hand, if a program is able to return the answer immediately, you can report that it takes “zero seconds”.

You are not required to turn in the softcopy of your program, the test-cases you have designed, or the running time for each individual test-case.