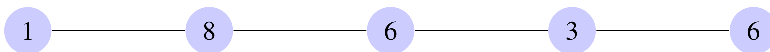
I collaborated with:

Problem
Answer Exercise 1 (page 312) in Chapter 6 of your text. For part (c) include a justification of correctness and a time/space complexity analysis.
Solution

A 

The algorithm will return the set of nodes {8,6}, while the independent set of maximum total weigth is {7,7,6}.

B 

The algorithm will return the set of nodes {1,6,6}, while the independent set of maximum total weigth is {8,6}.

C The recurrence $opt(v_i) = \max(w_i + opt(v_{i-2}), opt(v_{i-1})$ tells us a dynamic programming approach for solving this problem.

Algorithm 1 returns an independent set of maximum total weight

Start with A equal to the empty array of size n
function OPT(v)
  if v is adjacent to only one node of greater index or v is the last node **then**
    $A[i] = w_i$
    return $A[1]$
  else if $A[i] > 0$ **then**
    return $A[i]$
  else
    $A[i] = \max(w_i + opt(v_{i-2}), opt(v_{i-1}))$
    return $A[i]$
  end if
end function

This algorithm builds the array of maximum total weights for independent sets for all nodes from v_1 to v_n. We can then just find the maximum entry of A and trace subsequent maximum values of the remaining array to build up our answer. We will now prove why it works.

Proof. Base Case: Let g be a graph of only one node v_0. The independent set of maximum total weight is that only containing v_0. Our algorithm will see that v_0 is the last node of g and will update $A[0] = w_0$. It will then return $\{v_0\}$ as it corresponds to the maximum weight.
Inductive Hypothesis: Assume our algorithm works for all graphs that have up to k nodes where $k \geq 1$.
Inductive Step: We need to show our algorithm always returns an independent set of maximum total weight for a graph of $k+1$ nodes. Let G be a graph of k nodes labeled v_1 to v_k. If we add another node v_0 then we will need to see if v_0 is part of our independent set of maximum total weight. We will either include its weight to a maximum set of vertices not adjacent to v_0 (Note that this existing set corresponds to a graph of less than $k+1$ nodes and our inductive hypothesis assures the set is of maximum weights). Otherwise, we do not consider w_0 in our maximum set and we use the independent set of maximum total weight of G (by our inductive hypothesis, the algorithm returns the proper set for G as it has k nodes). □

Time Complexity: $O(n)$ to build array + $O(n)$ to find set = $O(n)$
Space Complexity: Array of size n $O(n)$ + n recursive calls for stack frame = $O(n)$