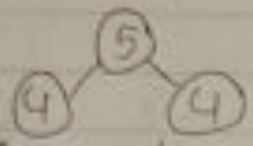
p1) a)



the "heaviest-first" algo would choose 5 then delete 4 and 4. This is not optimal because choosing 4 and 4 would lead to the independent set of maximum total weight

b)

The algorithm algo will choose the set $\{1,3\}$, but the set $\{2,3\}$ is the independent set with max total weight

c) $S(0)=0$ // Holds weight of max independent set of $V_{...}$

Recurse: $S(i)=\max(S(i-1), S(i-2)+w_i)$

use a table to hold already computed values for $S(i)$ at each step

Loop reverse: If $S(i)=S(i-1)$ then add v_{i-1} to A

where A is the answer ~~set~~ (in list form)

else $S(i)=S_i-2 \Rightarrow$ add v_{i-2} to A

return A

d) Proof: Induction

Base: With n = size of path, the array S[n] holds weight of max independent set

BC: For n=0 S[0]=0 because we initialize it to zero

IH: Assume S[m] is true for all $0 \le i \le n$

IS: Prove S[n+1] is correct

II cases:

1) $node_{n+1}$ is in the set when the first n+1 nodes are considered

2) $node_{n+1}$ is not in the set

1) If $node_{n+1}$ is in the set then $node_n$ is not in the set so we consider nodes, $node_1, \ldots node_{n-1}$ and we have S[n-1] as our MIS. We then add $node_{n+1}$ into the set and the new MIS is S[n+1]. Since n-1 < n+1 it is assumed that our algo computed the correct schedule and by adding a new disjoint node to the set, S[n+1] must also be correct.

2) If $node_{n+1}$ is not in the set then it must be the case that either S[n] or S[n-1] is the MIS and we have assumed these are correct so S[n+1] = (S[n] or S[n-1])

There are only two options, and the optimal option maximizes the weight for the schedule. This is what our algo computes, so it is

c) Runtime Analysis:

Going through the path takes $O(n)$ for n nodes

~~Cheating~~ Comparing values to find the max value is $O(1)$

Adding nodes to A takes $O(n)$ to add n nodes from $S[\_]$

Initializing $S[0] \Rightarrow O(1)$

~~$O(n) + O(n)$~~

$O(1) + O(n) * O(1) + O(n)$

$O(1) - O(n) + O(n)$

$O(n)$

Comparing values is nested inside of the recursive loop to compare values so we have

$O(n) * O(1)$