Dynamic Programming

Algorithms: Design and Analysis, Part II

WIS in Path Graphs: A Reconstruction Algorithm

Optimal Value vs. Optimal Solution

Recall: $A[0] = 0, A[1] = w_1$, for $i = 2, 3, \ldots, n$, $A[i] := \max\{A[i-1], A[i-2] + w_i\}$.

Note: Algorithm computes the <u>value</u> of a max-weight IS, not such an IS itself.

Correct but not ideal: Store optimal IS of each G_i in the array in addition to its value.

Better: Trace back through filled-in array to reconstruct optimal solution.

Key point: We know that a vertex v_i belongs to a max-weight IS of $G_i \iff w_i$ + max-weight IS of $G_{i-2} \geq$ max-weight IS of G_{i-1}.
(Follows from correctness of our algorithm!)

A Reconstruction Algorithm

Let A = filled-in array:

0	4	4	7	...	184
0	1	2	3	...	n

- Let $S = \emptyset$
- While $i \geq 1$ [scan through array from right to left]
  - If $A[i-1] \geq A[i-2] + w_i$ [i.e. case 1 wins]
    - Decrease i by 1
  - Else [i.e., case 2 wins]
    - Add v_i to S, decrease i by 2
- Return S

Claim: [By induction + our case analysis] Final output S is a max-weight IS of G.

Running time: $O(n)$