Dynamic Programming

Introduction, and WIS in Path Graphs

Algorithms: Design and Analysis, Part II

Problem Statement

Input: a path graph $G = (V, E)$ with nonnegative weights on vertices.

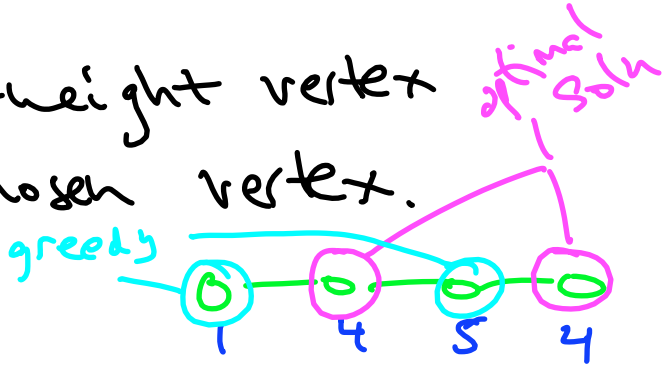
Desired output: subset of nonadjacent vertices — an independent set — of maximum total weight.

Next: iterate through our algorithm design principles.

Brute-force search: requires exponential time.

A Greedy Approach

Greedy: iteratively choose the max-weight vertex not adjacent to any previously chosen vertex.



Question: in example, what is value of the max-weight independent set, and that of the output of our greedy algorithm?

(A) 14 and 10

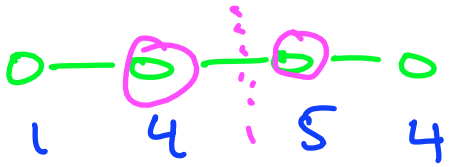
(B) 8 and 6

(C) 8 and 8

(D) 9 and 8

A Divide & Conquer Approach

Idea: recursively compute the max-weight IS of 1st half, ditto for 2nd half, then combine the solutions.



Problem: what if recursive sub-solutions conflict?

⇒ not clear how to quickly fix