

Homework #2: Dynamic Programming
Due Date: see Canvas

Assignment

Turn in solutions to **FOUR** of the problems below. Try to do some of the later problems, if you can.

Problem 1. Textbook Problem 15-3 (page 405, Bitonic Euclidean TSP). Clearly present your subproblems, how you solve a subproblem, and the time per subproblem. You probably also need to say something about backtracking (to recover the tour).

Problem 2. Textbook Problem 15-7 (page 408, Viterbi, both parts). Again: clearly present your subproblems, how you solve a subproblem, and the time per subproblem. You probably need to say something about backtracking (to recover the path).

Problem 3. Suppose G is a graph with tree decomposition T . Suppose C is a clique. That is, C a subset of the vertices of G such that every pair of vertices in C has an edge. Argue that some bag of T contains C .

Problem 4. Suppose you are given a graph G with n vertices, and a tree decomposition T of G , of treewidth at most k . That means each bag X_i in G has size at most $k + 1$.

4(a). Show how to modify T , so that no bag is a subset of an adjacent bag, and it still has treewidth at most k . (Hint: argue you can merge bags until you get the property.)

4(b). Argue that if you have the previous property, then T has $O(n)$ nodes. (Hint: root T , and argue that every node is the “topmost appearance” for some vertex of G .)

4(c). Show how to further modify T so that it is “smooth” (as defined on wikipedia). That is, each bag has size $k + 1$, and each pair of adjacent bags have k vertices in common.

Problem 5. Given a graph G with n vertices, and given a tree decomposition T for G of width k , we sketched in class an algorithm solving the weighted MIS problem in time $O(2^k \cdot k^c \cdot n)$. (The c is some constant.) How would you modify our algorithm to solve the weighted vertex cover (VC) problem, instead? (That is, to find a minimum weight subset C of vertices, so that every edge of G touches at least one vertex in C .) Try to argue a similar time bound for your algorithm.

Problem 6. Read the wikipedia page titled “Baker’s technique”. It sketches an algorithm (in fact a PTAS) that approximately solves the MIS problem in planar graphs. How would you modify that algorithm, to approximately solve the VC problem in planar graphs? In particular, how do you choose k (in terms of ϵ), what are your subgraphs of bounded treewidth (where VC can be solved exactly), and how do you combine those subproblems to get your final vertex cover? You should argue that your final vertex cover has size at most $1 + \epsilon$ times optimal.