

CS 170 HW 10

Due 2019-11-09, at 10:00 pm

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

In addition, we would like to share correct student solutions that are well-written with the class after each homework. Are you okay with your correct solutions being used for this purpose? Answer “Yes”, “Yes but anonymously”, or “No”

2 Exam Re-Attempt

For this problem, please re-attempt the midterm problems that you either did not answer (including the options for Questions 3 and 4 that you did not attempt) or did not get the right answer for on your first attempt.

After attempting each problem, read the solution. If you got it wrong, after reading the solution, close the solutions and summarize what you understood about the answer.

3 Graph Coloring Problem

An undirected graph $G = (V, E)$ is k -colorable if we can assign every vertex a color from the set $1, \dots, k$, such that no two adjacent vertices have the same color. In the k -coloring problem, we are given a graph G and want to output “Yes” if it is k -colorable and “No” otherwise.

- (a) Show how to reduce the 2-coloring problem to the 3-coloring problem. That is, given a graph G , show how to construct a new graph G' such that:
- If G is 2-colorable, G' is 3-colorable.
 - If G' is 3-colorable, G is 2-colorable.

In addition to giving the construction of G' , show that each of these properties holds in your solution.

- (b) The 2-coloring problem has a $O(|V| + |E|)$ -time algorithm. Does the above reduction imply an efficient algorithm for the 3-coloring problem? If yes, what is the runtime of the resulting algorithm? If no, justify your answer.

Solution:

- (a) To construct G' from G , we add a new vertex v^* to G , and connect v^* to all other vertices. If G is 2-colorable, one can take a 2-coloring of G and assign color 3 to v^* to get a 3-coloring of G' . Hence G' is 3-colorable.

If G' is 3-colorable, since v^* is adjacent to every other vertex in the graph, in any valid 3-coloring v^* must be the only vertex of its color. This means all remaining vertices only use 2 colors total, i.e. G is 2-colorable. We can recover the 2-coloring of G from the 3-coloring of G' by just using the 3-coloring of G' , ignoring v^* (and remapping the colors in the 3-coloring except for v^* 's color to colors 1 and 2).

- (b) No, the reduction is in the wrong direction. This reduction shows 3-coloring is at least as hard as 2-coloring, but it could be much harder.

(Comment: Indeed, it is suspected that there is no 3-coloring algorithm running in time $O(2^{|V|^{.99}})$. This is part of a common and somewhat mystical trend we will see more examples of very soon: lots of problems go from easy to hard when we change a 2 to a 3 in the problem description.)

4 Some Sums

Given an array $A = [a_1, a_2, \dots, a_n]$ of nonnegative integers, consider the following problems:

- 1 **Partition:** Determine whether there is a subset $P \subseteq [n]$ ($[n] := \{1, 2, \dots, n\}$) such that $\sum_{i \in P} a_i = \sum_{j \in [n] \setminus P} a_j$
- 2 **Subset Sum:** Given some integer t , determine whether there is a subset $P \subseteq [n]$ such that $\sum_{i \in P} a_i = t$
- 3 **Knapsack:** Given some set of items each with weight w_i and value v_i , and fixed numbers W and V , determine whether there is some subset $P \subseteq [n]$ such that $\sum_{i \in P} w_i \leq W$ and $\sum_{i \in P} v_i \geq V$

For each of the following clearly describe your reduction and justify its correctness.

- (a) Find a linear time reduction from SUBSET SUM to PARTITION.
- (b) Find a linear time reduction from SUBSET SUM to KNAPSACK.

Solution:

- (a) Suppose we are given some A with target sum t . Let s be the sum of all elements in A . If $s - 2t \geq 0$, generate a new set $A' = A \cup \{s - 2t\}$. If A' can be partitioned, then there is a subset of A that sums to t .

We know that the two sets in our partition must each sum to $s - t$ since the sum of all elements will be $2s - 2t$. One of these sets, must contain the element $s - 2t$. Thus the remaining elements in this set sum to t .

If $s - 2t \leq 0$, generate a new set $A' = A \cup \{2t - s\}$. If A' can be partitioned, then there is a subset of A that sums to t .

We know that the two sets in our partition must each sum to t since the sum of all elements will be $2t$. The set that does not contain $\{2t - s\}$ will be our solution to subset sum.

- (b) Suppose we are given some set A with target sum t . For each element k of the set, create an item with weight k and value k . Let $V = t$ and $W = t$. We know Knapsack will determine if there is a combination of items with sum of weights $\leq t$ and values $\geq t$. Because the weights and values are the same, we know (Sum of chosen weights) = (Sum of chosen values) = t . And since each weight/value pair is exactly the value of one of the original elements of A , we know that there will be a solution to our Knapsack problem iff there is one for our subset sum problem.