Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)

Homework C (*12/15* pts)

[10 pts] Consider the “equal partition” problem: Given a set of numbers, find out whether it is possible to partition the set of numbers into two subsets such that each set has the same sum, and that sum is half the total sum of the set of numbers.

For instance, the following set of numbers satisfies the equal partition problem: {1, 2, 3, 6}

The following set of numbers does not satisfy the equal partition problem: {1, 2, 3, 7, 8}

The goal of this problem is for you to prove that the equal partition problem is NP-Complete. I lead you through this step by step during the next two homeworks.

Recall the formal definition of the SUBSET-SUM problem: SUBSET-SUM = (<S,t> : there is a subset S’ ⊆ S such that t = Σs∈S’ s}

Formal definition: EQ-P = (<S,t> : there is a subset S’ ⊆ S such that t = 0.5\*(Σs∈S s)}

1. *1/2* Show that EQ-P is in NP. That is, suppose I gave you a set of numbers and two subsets, and I claimed that the union of these two subsets was identical to the original set and that each subset added up to half the sum of the original set. Explain in a couple of sentences how you would verify this, and why that verification could be done in polynomial time.[[1]](#footnote-1)

Answer: EQ-P is in NP: Guess the two partitions and verify that the two have equal sums. First, for every element a in A and b in B, verify that all the elements belonging to S are covered. Second, let S1 is 0 and S2 is 0. Thirst, For every element a in A add that value to S1. Fourth, for every element b in B add that value to S2. Finally, verify that S1 is the same as S2. So we can see that the algorithm could be done in polynomial time.

*Saying “… we can see that the algorithm could be done in polynomial time” isn’t the same as showing it.*

1. [5/5 pts]: Show how to convert any instance of the subset sum problem into an instance of the EQ-P problem. Specifically, given an instance of SUBSET-SUM = (<S,t> : there is a subset S’ ⊆ S such that t = Σs∈S’ s}, show how to convert this into an instance of the EQ-P problem EQ-P = (<U,x> : there is a subset U’ ⊆ U such that x = 0.5\*(Σu∈U u)}. Hint: Given sets S and S’ and sum t, explain how to choose sets U and U’ and sum x so that parts #2 and #3 of this problem are satisfied.

*Comment: Note that in order to choose the mapping for this part of the problem, I had to “think ahead” in order to identify a mapping that would meet the following two parts of the problem as well.*

Answer: recall SUBSET-SUM = (<S,t) : there is a subset S’ in S such that t = Σs∈S’ s}. Let x be the sum of members of S. Feed S’ = S ꓴ{x - 2t} into EQ-P. Accept if and only if EQ-P accepts. This reduction clearly works in polynomial time. So if there exists a set of numbers in S that sum to t, then the remaining numbers in S sum to x - t. therefore, there exists a partition of S’ into two such that each partition sums to x - t. Then, we get thatL let’s say that there exists a partition of S’ into two sets such that the sum over each set is x -t. One of there sets contains the number x - 2t. Removing this number, we get a set of numbers whose sum is t, and all of these numbers are in S.

1. [*3/5* pts]: Show that if there is a solution to the SUBSET-SUM problem, you can use that to prove the existence of a solution to the associated EQ-P problem.

Answer: We can consider the SUBSET-SUM problem can solve as a special case of the knapsack problem arising when the profit and the weight associated with each item are identical. The problem has numerous applications: Solutions of subset problems can be used for designing better lower bounds for scheduling problems. The constraints of 0-1 integer programs could be tightened by solving SUBSET-SUM problems with some additional constraints and it appears as subproblem in numerous combinatorial problems. So because of SUBSET-SUM problem ⊆ to EQ-P problem, so we can use the way of solving problem like knapsack problem to solve EQ-P broblem.

*You didn’t exactly answer the question. For #3 and #4, you pull up some interesting resources, but you neve really explain clearly what’s going on,*

1. [3/5 pts]: Show that if there is a solution to the EQ-P problem, you can use that to prove the existence of a solution to the associated SUBSET-SUM problem.

Answer: there is a solution to solve the EQ-P like the Complete Greedy Algorithm. This algorithm finds first the solution found by greedy number partitioning. It is fully polynomial-time approximation schemes for the SUBSET-SUM problem, and hence for the partition problem as well.

References:

* <https://en.wikipedia.org/wiki/Subset_sum_problem>
* <https://en.wikipedia.org/wiki/Partition_problem>
* <https://www.cs.cmu.edu/~ckingsf/class/02713-s13/lectures/lec15-subsetsum.pdf>
* <https://afteracademy.com/blog/partition-equal-subset-sum>
* <https://www.geeksforgeeks.org/subset-sum-is-np-complete/>
* <https://www.geeksforgeeks.org/partition-problem-dp-18/>
* <https://leetcode.com/problems/partition-equal-subset-sum/>

1. This is not a trick question, it should be pretty easy. Note that I am asking for two things: 1. Tell me how you would verify that the subset T sums to m (incredibly easy), and tell me how this is doable in time polynomial in the size of the set |S|. [↑](#footnote-ref-1)