Homework C (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. 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)

***Solution:***

This is easy to verify. I am given a list of numbers, and it is claimed that these numbers are in the set S and sum to t.

First, I verify that each number is in set S. If there are n numbers in the subset S’ that sums to t, I loop through the set S n times, and when I find the number in S, I delete it from the set S. This is at most O(n) work done n times, so this is an O(n2) algorithm.

It’s easy to sum the numbers in polynomial time and verify that their sum is t. that’s just basic addition, which is known to be a polynomial time operation.

1. [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.

***Solution:***

There are really two cases. If t = 0.5 \* T, then this is already the EQ-P problem.

Assume without loss of generality that t > 0.5 \* T. (If not, solve SUBSET-SUM<S,T-t>, which will be true if and only if SUBSET-SUM(<S,t>) is true.)

Then for the equivalent EQ-P problem, define v = T-t, and let U = S ∪ v. Let x = t. This procedure converts the subset sum problem into a version of the EQ-P problem. Because it only involves basic arithmetic, it’s clearly a polynomial time conversion.

***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.

1. [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.

***Solution:***

Suppose there is a solution to SUBSET-SUM(<S,t>). Then there is a solution to the EQ-P problem with U = S ∪ v, because the sum of all numbers in U is T + v = t + (T-t) + v = T + v.

1. [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.

***Solution:***

Suppose there is a solution to EQ-P(<U,t+v>). Since v is one of the numbers in set U, this means that there is a set of numbers in U – v = S which sums to (t+v)-v = t, so SUBSET-SUM(<S,t>) is true.

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)