Homework 19 sample solution

Due 11/23/16

November 17, 2016

1. The `Subset Sum Decision` problem (SSD) accepts an array of integers *data* and target t and returns whether *data* contains some subset that sums up to t. Prove that $SSD \in NP$.

   You can either show that SSD is poly-time verifiable, or that it has a nondeterministic poly-time algorithm.

   A poly-time verification algorithm would accept the solution S, in addition to *data* and t. It should test that $S \subseteq data$ and S sums to t. The first can be accomplished in $\Theta(n)$ expected time through the use of a hash table, and the second takes $O(|S|)$ time, which is $O(n)$, for a total of $\Theta(n)$ expected case time.

   A nondeterministic poly-time algorithm can simultaneously compute the sum of every subset of *data*, returning true if one sums up to t. This takes $O(n)$ nondeterministic time, as this is how long it takes to add up the subset containing all of *data*.

2. The `Subset Sum` problem (SS) accepts the same input as SSD, but it returns a subset that sums to t rather than true or false. The following algorithm describes a reduction from SS to SSD:

```
   Input: data: array of integers
   Input: n: size of data
   Input: t: target value
   Output: S: subset of data that sums to t, or ∅ if no such set exists
 1 Algorithm: SSReduction
 2 if ¬SSD(data, t) then
 3 |  return ∅
 4 end
 5 sub = 0
 6 top = n
 7 while sub < top do
 8 |  if ¬SSD(data[1..(top − 1)], t) then
   |  |  /* Add data[top] to subset                                   */
 9 |  |  sub = sub + 1
10 |  |  Swap data[top] and data[sub]
11 |  else
   |  |  /* Delete data[top] from array                               */
12 |  |  top = top − 1
13 |  end
14 end
15 return data[1..sub]
```

What do we know about the worst-case complexity of SS if the worst-case complexity of SSD is $O(S(n))$ and $\Omega(s(n))$?

Since each iteration of the while loop increments *sub* or decrements *top*, it will iterate exactly n times, for a total complexity of $O(nS(n))$ time, which dominates the cost of the other lines. Since SSReduction takes $O(nS(n))$ time, SS must be $O(nS(n))$.