

## Solution Set 6

Posted: March 7

1. (a) First, note that PARTITION is in NP because given subset  $T \subseteq S$  we can verify in polynomial time that  $\sum_{a \in T} a = \sum_{a \in S-T} a$ .

To show that PARTITION is NP-hard, we reduce from SUBSET SUM. Given an instance  $(S = \{a_1, a_2, \dots, a_n\}, B)$  of SUBSET SUM, let  $M = \sum_i a_i$ . Our reduction produces the following instance of PARTITION:

$$S' = S \cup \{p = L - B, q = L - (M - B)\},$$

where  $L = M + 1$ . Clearly this reduction runs in polynomial time.

If we started with a YES instance of SUBSET SUM, then we claim that the reduction produces a YES instance of PARTITION. Suppose there exists a subset  $T \subseteq S$  for which  $\sum_{a \in T} a = B$ . Then we have  $\sum_{a \in S-T} a = M - B$ , and so we have  $p + \sum_{a \in T} a = L = q + \sum_{a \in S-T} a$ , which implies that  $S'$  is partitionable.

If  $S'$  is a YES instance of PARTITION, then we claim that  $(S, B)$  is a YES instance of SUBSET SUM. Let  $T' \subseteq S'$  specify the partition. Observe that we can't have both  $p$  and  $q$  in the same part of the partition, because then the sum of the integers in that part would be at least  $p + q = 2L - M > M$ , and the sum of the integers in the other part would be at most  $M$ . The sum of all elements in  $S'$  is  $2L$ , so we must have:

$$\sum_{a \in T'} a = L = \sum_{a \in S'-T'} a.$$

If  $p$  is in the first part, then  $T' - \{p\}$  is a subset of elements of  $S$  that sum to  $B$ , and if  $p$  is in the second part, then  $(S' - T') - \{p\}$  is a subset of elements of  $S$  that sum to  $B$ . We conclude that  $(S, B)$  is a YES instance of SUBSET SUM as required.

- (b) First, note that KNAPSACK is in NP because given subset of the  $n$  elements, we can verify in polynomial time that the sum of their values is at least  $V$ , and the sum of their costs is at most  $C$ .

To show that KNAPSACK is NP-hard, we reduce from SUBSET SUM. Given an instance  $(S = \{a_1, a_2, \dots, a_n\}, B)$  of SUBSET SUM, our reduction produces the following instance of KNAPSACK: the cost  $c_i$  of item  $i$  is set to  $a_i$ , and the value  $v_i$  of item  $i$  is set to  $a_i$  as well. We set  $V = C = B$ . Clearly this reduction runs in polynomial time.

If we started with a YES instance of SUBSET SUM, then we claim that the reduction produces a YES instance of KNAPSACK. Suppose there exists a subset  $T \subseteq S$  for which  $\sum_{a \in T} a = B$ . Then packing the element in  $T$  into our knapsack costs  $B$  and has value  $B$ , so the instance of KNAPSACK produced by the reduction is a YES instance.

If the reduction produces a YES instance of KNAPSACK, then we claim that  $(S, B)$  is a YES instance of SUBSET SUM. Let  $T \subseteq S$  be the items packed into the knapsack,

whose total value is at least  $V$  and whose total cost is at most  $C$ . In other words  $\sum_{a \in T} a \geq V = B$  and  $\sum_{a \in T} a \leq C$ , which implies that  $\sum_{a \in T} a = B$ . We conclude that  $(S, B)$  is a YES instance of SUBSET SUM as required.

- Following the hint, we observe that in an  $n$ -node graph  $G = (V, E)$ , there is a bisection with *at least*  $k$  edges crossing it iff there is a bisection with *at most*  $n^2 - k$  edges crossing it in the complement graph  $G' = (V, \bar{E})$ . Since both MINIMUM BISECTION and MAXIMUM BISECTION are in NP, and this observation yields a simple reduction from MAXIMUM BISECTION to MINIMUM BISECTION (namely, map the instance  $\langle G = (V, E), k \rangle$  of MAXIMUM BISECTION to the instance  $\langle G' = (V, \bar{E}), |V|^2 - k \rangle$ ), we need only show that MAXIMUM BISECTION is NP-hard.

Here, we reduce from MAX CUT, which was shown to be NP-complete in class. The reduction is very simple. Given an instance  $\langle G = (V, E), k \rangle$  of MAX CUT, we produce the graph  $G'$  which is just  $G$  with  $|V|$  additional, isolated nodes, and  $\langle G', k \rangle$  is our instance of MAXIMUM BISECTION. Clearly this reduction runs in polynomial time. We now argue that yes maps to yes and no maps to no. If  $G$  has a cut  $S \subseteq V$  with at least  $k$  edges crossing it, then if we add  $|V| - |S|$  of the isolated nodes to that cut, we obtain a bisection of  $G'$  with at least  $k$  edges crossing it (so yes maps to yes). If  $G' = (V', E)$  has a bisection  $S \subseteq V'$  with at least  $k$  edges crossing it, then if we simply discard the isolated nodes from that bisection, we obtain a cut in  $G$  with at least  $k$  edges crossing it (so no maps to no).

- We want to show that MINIMUM EQUIVALENT CIRCUIT is coNP-hard. We reduce from DNF TAUTOLOGY (from class). Let  $\phi$  be an instance of DNF TAUTOLOGY. First, check if  $\phi \equiv 0$ . Since  $\phi$  is a DNF formula, this happens iff *every* term is unsatisfiable (i.e. contains both a variable and its negation). Therefore, this can be checked in polynomial time by examining each term. If this check shows that  $\phi \equiv 0$ , then our reduction produces a fixed negative instance of MINIMUM EQUIVALENT CIRCUIT — say,  $(x \vee y, 1)$ .

Otherwise, we know that  $\phi \not\equiv 0$ , and our reduction produces the instance  $(C = z \wedge \phi, 1)$  of MINIMUM EQUIVALENT CIRCUIT. Here,  $z$  is a fresh variable. Now if  $\phi \equiv 1$ , then  $z \wedge \phi \equiv z$  and so there is an equivalent circuit of size 1. In the other direction, if  $\phi \not\equiv 1$ , then we claim that the smallest circuit possible has size 2. It is clear that  $(z \wedge \phi) \not\equiv 0$  since we have ensured that  $\phi \not\equiv 0$ . Similarly, it is clear that  $(z \wedge \phi) \not\equiv 1$ . Finally,  $(z \wedge \phi)$  cannot be equivalent to a circuit of size 1, because then the equivalent circuit would consist of a single variable. If this variable was  $z$  then  $\phi \equiv 1$  (contrary to our assumption); and we cannot have  $x_i \equiv (z \wedge \phi)$  because  $(z \wedge \phi)$  is false when  $z$  is false and  $x_i$  is true, but  $x_i$  is true under this assignment.

You were asked to think about whether this problem was in coNP (you did not need to write anything here). The belief is that MINIMUM EQUIVALENT CIRCUIT is *not* in coNP, but no one knows how to prove this. Here is a rough idea of why it seems not to be in coNP. If it were in coNP, then the complement language

$$\overline{\text{MINIMUM EQUIVALENT CIRCUIT}} = \{(C, k) : \text{for all } C' \text{ with } |C'| \leq k, C' \not\equiv C.\}$$

would be in NP. However, there doesn't seem to be any short certificate of membership here; it seems that the only sort of certificate is to list *all*  $C'$  of size at most  $k$ , together with an  $x$  such that  $C'(x) \neq C(x)$  for each. In fact the “for all” seems to suggest that the *complement* is coNP-hard (and indeed this is believed to be the case).

4. We use as our starting point the fact that PRIMES is in NP. This means that PRIMES can be expressed as

$$\text{PRIMES} = \{x : \exists y, |y| \leq |x|^k, (x, y) \in R\}$$

for some language  $R \in P$ . For prime  $x$ , for convenience, we will denote by  $C(x)$  a certificate of primality; i.e.,  $C(x) = y$  such that  $(x, y) \in R$  (we select arbitrarily from among possibly several such  $y$ ).

We first show that  $L$  is in NP. To do this we describe a succinct certificate of membership, that can be polynomially verified. A certificate that  $x \in L$  consists of the following components:

- a complete prime factorization of  $x$ :  $q_1, q_2, \dots, q_m$ .
- a certificate of primality for each  $q_i$ :  $C(q_1), C(q_2), \dots, C(q_m)$ ,
- a certificate of primality for the number  $\ell$  of distinct integers among  $q_1, q_2, \dots, q_m$ :  $C(\ell)$ .

Observe that this certificate has length polynomial in  $|x| = \log_2 x$ . This follows because  $x$  has at most  $\log_2 x$  factors in its prime factorization. Since each  $q_i \leq x$ ,  $|q_i| \leq \log_2 x$ . Finally, we know that  $|C(x)| \leq \log^k x$  for some  $k$ . Overall, the length of the certificate is at most  $\log^{k'} x$  for some  $k'$ .

To verify this certificate, we first check that  $q_1 q_2 \dots q_m = x$ . Then we check that  $(q_i, C(q_i)) \in R$  for all  $i$ . Finally we check that there are indeed  $\ell$  distinct integers among  $q_1, q_2, \dots, q_m$ , and that  $(\ell, C(\ell)) \in R$ . This takes polynomial time overall (since  $R \in P$ ), and in the end we accept iff there are a prime number of distinct number of prime divisors of  $x$ .

Next we show that  $L$  is in coNP. To do this we describe a succinct certificate of non-membership, that can be polynomially verified. A certificate that  $x \notin L$  consists of the following components:

- a complete prime factorization of  $x$ :  $q_1, q_2, \dots, q_m$ .
- a certificate of primality for each  $q_i$ :  $C(q_1), C(q_2), \dots, C(q_m)$ ,
- a proper divisor of the number  $\ell$  of distinct integers among  $q_1, q_2, \dots, q_m$ :  $d$

This certificate has length at most  $\log^{k'} x$  for the same reasons that the previous one does.

To verify this certificate, we first check that  $q_1 q_2 \dots q_m = x$ . Then we check that  $(q_i, C(q_i)) \in R$  for all  $i$ . Finally we check that there are indeed  $\ell$  distinct integers among  $q_1, q_2, \dots, q_m$ , and that  $d$  divides  $\ell$ . This takes polynomial time overall (since  $R \in P$ ), and in the end we accept iff there are a composite number of distinct number of prime divisors of  $x$ .