02249 - Computationally Hard Problems, Assignment IV

Arianna Taormina, S163671

28 September 2020

Exercise Description

[Simple-3-Partition] Input: Given are n natural numbers $s1, s2, ..., s_n$.
Output: YES if the numbers can be partitioned into three sets with equal sums, i. e., if there are disjoint sets $A_1, A_2, A_3 \subseteq 1, ..., n$ such that $A_1 \cup A_2 \cup A_3 = 1, ..., n$ and

$$\sum_{i \in A_1} s_i = \sum_{j \in A_2} s_j = \sum_{k \in A_3} s_k$$

and NO otherwise.
Prove that Simple-3-Partition is in the class NP.

Exercise Resolution

1) Let $s1, s2, ..., s_n$. be the sequence of n natural numbers

a) The random string R consists of random sequence of integers of length n, where each integer is either 1,2 or 3.

$$R = r_1 r_2 ... r_n$$

$$\text{with } 1 \leq r_i \leq 3 \text{ for } i = 1...n$$

b) The algorithm A takes R as an assignation guess and swiping from left to right bins each s_i in the correspondent guessed set.

$$\text{if } r_i = 1 \text{ then } s_i \in A_1$$

$$\text{if } r_i = 2 \text{ then } s_i \in A_2$$

$$\text{if } r_i = 3 \text{ then } s_i \in A_3$$

c) A computes the sum of s_i in each of the three sets and compares them. If they are equal to each other then returns YES, otherwise returns NO.

2) We show now that the two conditions of NP problems are met:

a) "If the answer to X is YES, then there is a string R0 with positive probability such that A(X, R0) = YES." If the simple tri-partition exists, it can be represented by a string of numbers in range 1 to 3, of length n, for example:

$$S = 3, 6, 1, 5, 3$$

$$A_1 = \{1, 5\}; A_2 = \{3, 3\}; A_3 = \{6\}$$

$$R0 = 2, 3, 1, 1, 2$$

For a sequence of length n, if every integer can have only 3 values, there are 3^n possible combinations, among which, if exists, is R0. So the probability of R0 to be generated by a uniform random choice from this set of sequences is small but positive.

b)"If the answer to X is NO, then A(X,R) = NO for all R." If the simple tri-partition doesn't exists, none of the possible sequences will return a YES. The algorithm A will always output NO in this case, whence the second condition is satisfied.

3) In the first step for each $i = 1...n$ the algorithm looks at r_i value, from a finite set (in this case 3 values, anyhow it is search-able in constant time) and assigns consequentially s_i to one of the three disjoint sets. Time complexity O(1)*O(n) = O(n). Then for each subset the algorithm sums over the elements and finally compares the three sums. Time complexity O(n)+O(1) = O(n). Hence the overall time complexity of the algorithm is O(n).