Opracowanie: e/
Problem E: Bit Compressor

HISTORIA:
• wersja 1.1: 2007, Tomasz Idziaszek
• wersja 1.0: 2007, Karolina Sołtys, Marcin Balcerzak

dokument systemu SINOL 1.3.1

1 Definitions

Let us first introduce some terms used later on. From now on by subsequence we understand a connected subsequence.

Definition 1 Coding *is a compression as it is defined in the content of the problem.*

Definition 2 Decoding *is a decompression — an opposite process to compression.*

Definition 3 Output sequence *is a 0–1 sequence that is an input data to the program to decode.*

Definition 4 Input (decoded) sequence *is a sequence that has appropriate length, number of 1's and after encoding gives the output sequence.*

The aim is to check whether for input data the number of possible input sequences is 0, 1 or greater than 1. Then we correspondingly return: NO, YES, NOT UNIQUE.

Definition 5 A block *of the given sequence is the subsequence of the form* $1 \ldots 10 \ldots 0$ *such that there is no bigger subsequence that contains it.*

2 Observations

Below we make few observations that will be considered as obvious later on.

Fact 1 *Decoded subsequences of 1's don't touch each other.*

Fact 2 *The subsequence of two 1's that don't touch the decoded subsequences of 1's from the rest of the output sequence can stem both from* 11 *and* 111*.*

Fact 3 *Subsequences in the output sequence that code subsequences of 1's in input sequence begin with* 1 *not preceded by* 1 *and end with* 1 *preceding* 0 *(or ending the sequence) or* 0 *not preceding* 1*.*

3 The general concept

When given the length of the input sequence, the number of 1's in it and the output sequence we know the number of 0's that emerged because of coding. The solution bases on checking possible divisions of this quota to 0's in the output sequence and counting the divisions correct according to the content of the problem.

4 Implementation

We keep a table (of the maximum length of 21) of elements that describe blocks and concatenation of blocks. The description contains the number of 1's acquired from decoding the block 9the initial value don't take into account the 0's in the block) and the number of 0's that we can use in the block (initialized with the number of all 0's in the block).

We divide the quota of 0's stemming from the compression in a recurrent way, having the initial quota and starting from the first (leftmost) block. We notice that if the number of 0's in the block is not greater than the number of 0's in currently available quota to divide, we should concatenate two consecutive blocks (the subsequent one has a modified description of the state) and resolve problem for this subsequent block — to not to pass over some solution. Afterwards we do also alloting the quota to the block and resolving the problem for the next block.

The number of found solutions and the number of 1's acquired in the process of encoding the part of sequence on the left of currently processed block are remebered globally.

The solution is correct when after the last block we are out of qouta of 0's, the length of the input sequence is correct and its number of 1's also.

5 Correctness

5.1 Stop condition

An algorithm stops after processing all blocks and the number of possible divisions of 0's between blocks is finite, thus the algorithm stops.

5.2 Partial correctness

Found solutions are correct. We don't miss any decoding because the searching is complete.

6 Complexity

The memory complexity is constant while there is no dynamic allocation of the memory. Maximum pessimistic depth of recurrent calls of the function processing blocks is 21. Pessimistic time complexity is exponential with reference to the product of the lengths of all maximum subsequences of 0's increased by 1. This product is highest when the subsequences have the same length so for the n subsequences it is estimated by $(\frac{40-n}{n}+1)^n$, that is always less than 2^{22} (maximum for $n = 15$).

7 Tests

The intrinsic part of the solution are test sets. Below there is a short description:

1.in checks the border cases for small input data

2.in correctness tests for output sequences of the length of 5

3.in correctness tests for output sequences of the length of 10

4.in correctness and efficiency tests for output sequences of the length of 15

5.in correctness and efficiency tests for output sequences of the length of 20

6.in correctness and efficiency tests for output sequences of the length of 30

7.in correctness tests for random output sequences of the length of 40

8.in correctness tests for random output sequences of the length of 40

9.in correctness tests for random output sequences of the length of 30

10.in correctness tests for random output sequences of the length of 20