PROBLEM 5: STRANGE NETWORK

Team: *Russia.*

ABSTRACT:

Part 1 of the problem was completely solved. Following facts were obtained during research:

- Following the problem statement, α shouldn't be less than $n-1$. There's no any other relation between α and T_{min}
- For $k=1$ problem is obvious: if $n=1$ then $T_{min}=1$, else code cannot be restored at all
- For $k \geq 2$ minimal number of k-subcodes, such as any code of length n can be restored from them is $T_{min}(n,k) = C_n^k - C_{n-2}^{k-2} + 1$

Part 2 is quite more difficult, but there also are some results for it:

- For $\alpha = 0$ it is easy to see that $T_{min} = 0$
- Even in case of $\alpha = 1$ for some k exists N such as for any $n > N$ code of length n cannot be restored by k-subcodes at all. Examples are given in this paper.
- Previous statement can also be applied for any $\alpha \geq 1$.
- If any code of length n can be restored from k-subcodes, then $T_{min}(n,k) \geq C_n^k - C_{n-2}^{k-2} + 1$.
- As it was noticed before, we made some examples of codes of length n that cannot be restored from any set of k-subcodes. So we have hypothesis about the way such an examples can be constructed for case of $n > 2^k$.

1. Following the definition of this part of problem, all numbers in code must be different, so α must be not less than $n-1$.

To begin with, let's consider case of $k=1$. If $n=1$ code will be restored by first k-subcode. If $n \geq 2$ it's easy to see that some n-codes cannot be restored from k-subcodes: for codes $(1,2,3,...,n)$ and $(2,1,3,...,n)$ there are same sets of 1-subcodes.

Case of $k \geq 2$ is completely solved too: $T_{min} = C_n^k - C_{n-2}^{k-2} + 1$. Let us show that this number is really minimal, so exists at least one code for each n that cannot be restored from $C_n^k - C_{n-2}^{k-2}$ k-subcodes. Codes $(0,1,2,3,...,n-2,n-1)$ and $(0,1,2,3,...,n-1,n-2)$ can give $C_n^k - C_{n-2}^{k-2}$ same subcodes: number of subcodes that contains last two numbers in code is C_{n-2}^{k-2}. So number of codes that doesn't contain both of this numbers simultaneously is $C_n^k - C_{n-2}^{k-2}$ and even if we get all of this subcodes, it will be impossible to restore code - there is still at least two variants. Now let's show that $C_n^k - C_{n-2}^{k-2} + 1$ of subcodes is enough: let's assume that code $(a_1,...,a_n)$ cannot be restored from such a number of k-subcodes. All the a_i are different, so we cannot restore code only if for some pair of a_i, a_j we cannot say what of them will be met in code earlier (it includes such a situation when some a_i is still unknown). So let's assume that we got $C_n^k - C_{n-2}^{k-2} + 1$ k-subcodes and still don't know for some i,j what of a_i, a_j will be met earlier. But number of k-subcodes that don't contain a_i and a_j simultaneously is $C_n^k - C_{n-2}^{k-2}$, so we have at least one subcode that contains this pair and gives us information about order of a_i, a_j in code, that contradicts our assumption. So $T_{min} = C_n^k - C_{n-2}^{k-2} + 1$ is really enough number of k-subcodes for restoring code in any case, and it is minimal among all numbers of subcodes that gives us guaranteed possibility of restoration code.

2. First, and the most obvious case is $\alpha = 0$. In such a situation no subcodes are needed to restore code: it is just $(0,0,...,0)$ where length of code n, as defined, is already known.

But even for $\alpha = 1$ there is many different examples of n, k and code that cannot be restored at all: the easiest one is for $k=1, n=2$ - codes $(1,0)$ and $(0,1)$ gives the same set of 1-subcodes. It is trivial example but there is also some of them for larger k and n such as this one: codes $(1,0,0,1)$ and $(0,1,1,0)$ gives the same set of 2-subcodes, that means that even if we'll get all 2-subcodes we cannot differ this two codes. Now let's do some trick: append each of this codes to the end of other, getting $(1,0,0,1,0,1,1,0)$ and $(0,1,1,0,1,0,0,1)$. There is example for $k=3$ and $k=2$. By repeating such an operation we got an example for $k=2, k=3, k=4$; and when we repeated it once again we got example for k from 2 to 5. So our hypothesis is that if we continue such an operation we'll get examples for any k and $n = 2^k$.

Also, if we have such an example for some k and n than we can easily construct an example for k and any $N \geq n$: all we need is to append any numbers to both codes to get length of N.

It is easy to see that for any $\alpha \geq 1$ this examples can be applied without any change.

Let's assume that for some n, k and $\alpha \geq 1$ T_{min} is defined. It cannot be less than $C_n^k - C_{n-2}^{k-2} + 1$: for code $(0, 0, ..., 0, 1)$ we can get all k-subcodes that doesn't contain last pair $(0, 1)$. As it was shown before it is $C_n^k - C_{n-2}^{k-2}$, so we can have $C_n^k - C_{n-2}^{k-2}$ k-subcodes that won't give us enough information to differ $(0, 0, .., 0, 1)$ from $(0, 0, ..., 1, 0)$. Let's sum up: for several cases code cannot be restored in all and even if restoration is possible, T_{min} is close to number of all k-subcodes, so for large n and non-trivial k there is no any fast algorithm for restoration of code.