Algorithm Design Techniques

Optional Cool Problems I

10 January 2026

It is okay to be stupid. Wait, I meant it is okay to feel stupid.

— Prof. Aditya Karnataki

§1 Problems

1. **Spying with coins (IOI 2017 Practice)**

   You have a number C $(0 \leq C < 63)$ and an line of 64 coins that are either heads or tails.

   As an secret agent, you want to communicate your number to your handler by flipping some of the coins. To avoid being caught, you want to use as few flips as needed. To make the handler aware that you are communicating, you wish to flip atleast one coin. (The handler doesn't know the initial sequence of coins).

   In the encoding part, you may flip at least one coins and at most K coins of the line.

   In the decoding part, you receive the coins already with the changes, in a line, and you must recover the number C.

   Subtask 1: $K = 64$, This is easy.

   Subtask 2: $K = 6$, Using our friend binary.

   Subtask 3: $K = 1$, Note that bitwise Xor $\oplus$ (google it) has some very useful properties. One of these is the fact that it is extremely easy to change and second is that for numbers between $1 - 64$, taking bitwise xor of some set of numbers will result in a number between $0 - 63$. How can we abuse it?

2. **Talking with a Broken Device (JOI 2016, Spring Training Camp)**

   Anna wants to send a 60-bit integer to Bruno. She has a device that can send a sequence of 150 numbers that are either 0 or 1. The twist is that L $(0 \leq L \leq 40)$ of the positions of the device are broken and can only send 0. Bruno receives the sequence Anna sent, but the does not know the broken positions.

   Anna knows the broken positions but Bruno doesn't. Write functions `encode` which given the integer and the broken position encodes the message and function `decode` which decodes the message and recovers the integer sent.

   Subtask 1: $K = 0$, This should be very simple as you are just converting to binary.

   Subtask 2: $K = 1$, We will have one broken position. If we can somehow indicate the start of out 60 bit sequence, can we find a continuous 61 bit sequence?

   Subtask 3: $K = 15$, This is where one needs to be creative. Note $\frac{150}{2} = 75$ and $75 - 15 = 60$. Can you think of something now?

Subtask 4: $K = 40$, The last question had 2, now try with 3 but have some sequences encode more than 1 bit.

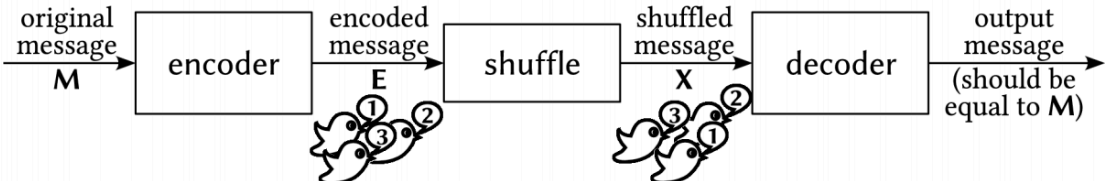
3. **Parrot Data Transfer Protocol (IOI 2011, P6)**

Yanee is a bird enthusiast. Since reading about IP over Avian Carriers (IPoAC), she has spent much of her time training a flock of intelligent parrots to carry messages over long distances.

Yanee's dream is to use her birds to send a message M to a land far far away. Her message M is asequence of N (not necessarily distinct) integers, each between 0 and 255, inclusive. Yanee keeps some K specially-trained parrots. All the parrots look the same; Yanee cannot tell them apart. Each bird can remember a single integer between 0 and R, inclusive.

Early on, she tried a simple scheme: to send a message, Yanee carefully let the birds out of the cage one by one. Before each bird soared into the air, she taught it a number from the message sequence in order. Unfortunately, this scheme did not work. Eventually, all the birds did arrive at the destination, but they did not necessarily arrive in the order in which they left. With this scheme, Yanee could recover all the numbers she sent, but she was unable to put them into the right order.

To realize her dream, Yanee will need a better scheme, and for that she needs your help.



Try to design a function that receives the **Original Message** (of some length N with numbers between $0 - 255$) and encode the original messages to another message sequence, it's called **Encoded Message** with limit the size of message must not exceed K and using numbers from $0 - R$. The **encoded message** from will be shuffled. Receive the **Shuffled Message** and Decode back to **Original Message**.

We suggest the following roadmap,

Subtask 1 : $N = 8$, all elements of the orignal message are either 0 or 1, $R = 2^{16} - 1$, $K = 10 * N = 80$.

Subtask 2 : $1 \leq N \leq 16$, $R = 2^{16} - 1$, $K = 10 * N$

Subtask 3: $1 \leq N \leq 16$, $R = 255$, $K = 10 * N$

Subtask 4: $1 \leq N \leq 32$, $R = 255$, $K = 10 * N$

Subtask 5: $1 \leq N \leq 32$, $R = 255$, $K = 10 * N$

Subtask 6: We now want you to try to reduce the ratio between encoded message length and original message length upto $1 \leq N \leq 64$. The mathematical limit for the best ratio one can get is slightly above $\frac{261}{64} \approx 4.08$ (Derive the limit!). Anything below 7 is very good, although we (the TAs) are very interested if someone can find the optimal solution.

Note: When the problem came in IOI, no one found the optimal solution. Furthermore, most solutions which got 100, did so they were optimal on the test cases and not in the general case.