IOI 2024 Solutions: Problem Message

Problem Information

Problem Author: Arthur Nascimento (Brazil)

Problem Preparation: Mahmoud Badawy, Bartosz Kostka, Mohamed Anany

Editorial: Masataka Yoneda, Hirotaka Yoneda

1. Subtask 1 (10 points)

In this subtask, consider sending one bit of message per packet. We can use the following strategy:

- **Aisha's Strategy:** If she wants to send `0`, she sends a packet with `000...000`. If she wants to send `1`, she sends a packet with `111...111`.
- **Basma's Strategy:** In the packet which Basma received, if (number of `0`'s) > (number of `1`'s), she knows that the message is `0`. Otherwise, she knows that the message is `1`.

This strategy works because, Cleopatra can only change at most 15 bits, which is less than half of the packet, so the correct bit remains the majority. In coding theory, this method is called **repetition code**.

2. Subtask 2: $Q = 96$ (29.31 points)

Basic Insights

First, consider for simplicity that the message length S always equals 1024.

To get some insights of the solution, thinking about some easier variation of the problem may work. This time, let's think about the following question: *"what if Basma knew precisely which* 15 *indices are controlled by Cleopatra?"* (we will call them 'controlled bits' from now on). In this hypothetical scenario, Aisha can send 16 bits of message per packet, by simply embedding the message in 16 non-controlled bits. Therefore, it only requires $Q = \lceil \frac{1024}{16} \rceil = 64$ packets. (Note: $\lceil x \rceil$ is the smallest integer which is at least x)

The problem is that Basma does not actually know the 15 non-controlled bits. One way to tackle the problem is to think about the strategy to convey the positions of 15 non-controlled bits, by using some packets.

The Solution when $S = 1024$

Aisha uses first 31 packets to tell which bits are non-controlled. Using the idea of Subtask 1, in $(i + 1)$-th packet $(0 \leq i \leq 30)$, Aisha sends `000...000` when $C_i = 0$, and Aisha sends `111...111` when $C_i = 1$. After completing this process, now Basma knows which bits are controlled. Then, Aisha can send the entire message in the next 64 packets using the strategy explained in **Basic Insights** section. Overall, it requires $Q = 31 + 64 = 95$ packets.

The Solution for General S

The actual problem is that $1 \leq S \leq 1024$ and Basma does not know the message length S beforehand. However, the action of sending a message of length between 1 and 1024, can be transformed to the action of sending a message of length **exactly 1025**. This is because, the number of species of messages is $2^1 + 2^2 + \cdots + 2^{1024}$, which is slightly less than 2^{1025}.

One way of implementing this is to first send `0`'s before the message, then a `1` to denote the beginning of the message, and finally the actual message, so that the total length is exactly 1025. For example, if the message is `101...100` with length 1019, we send the following string (underlined section is the actual message):

`000001`<u>`101...100`</u>

Back to the main problem, it requires $Q = 31 + \left\lceil \frac{1025}{16} \right\rceil = 96$ packets.

3. Subtask 2: $Q = 85$ (43.97 points)

The main focus on the next step is: *"how to convey which bits are controlled, in less than* 31 *packets?"*.

The problem is that we only sent 1 bit of information per packet, which may be inefficient. For example, one can notice that, if Basma *already* found 2 non-controlled bits, Aisha can send 2 bits of information by embedding it in these bits, which is more efficient. (Note: It can be compared to the latter phase of $Q = 96$ solution, when Basma already found 16 non-controlled bits and Aisha sends 16 bits of information.)

We consider the situation that Aisha already sent the information of $C_0, C_1, \ldots, C_{i-1}$, and k of them are 0 (non-controlled bits). Then instead of sending `000...000` or `111...111`, Aisha sends the next k values of the C_i's.

The worst case is that the first 15 bits are controlled bits ($C_0 = \cdots = C_{14} = 1$) and the last 16 bits are non-controlled bits ($C_{15} = \cdots = C_{30} = 0$). Even in this case, the following thing happens:

- Aisha sends the information of $C_0, \ldots, C_{16}$ in the first 17 packets.
- Aisha sends the information of C_{17}, C_{18} in the next packet.
- Aisha sends the information of $C_{19}, C_{20}, C_{21}, C_{22}$ in the next packet.
- Aisha sends the information of $C_{23}, \ldots, C_{30}$ in the next packet.

So the phase of conveying which bits are controlled is done in 20 packets. Back to the main problem, it requires $20 + 65 = 85$ queries.

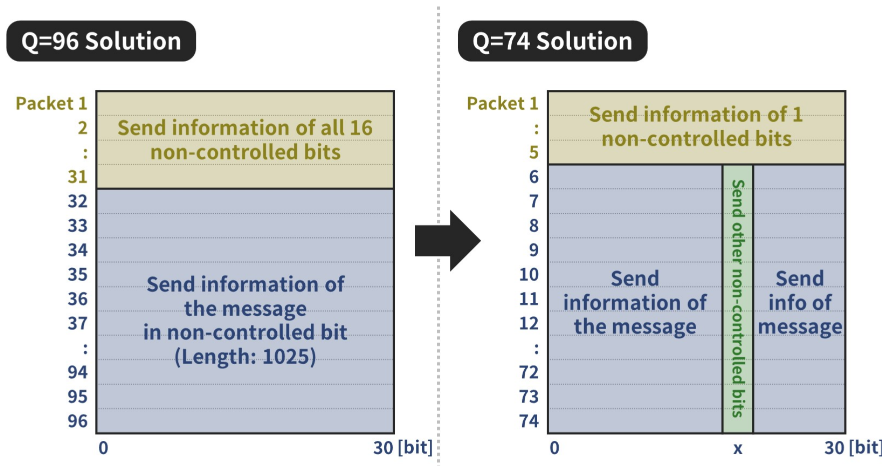
4. Subtask 2: $Q = 71$ (79.64 points)

For the solutions in Chapters 2 & 3, Aisha sends the information of all 16 non-controlled bits first and then sends the information of the message.

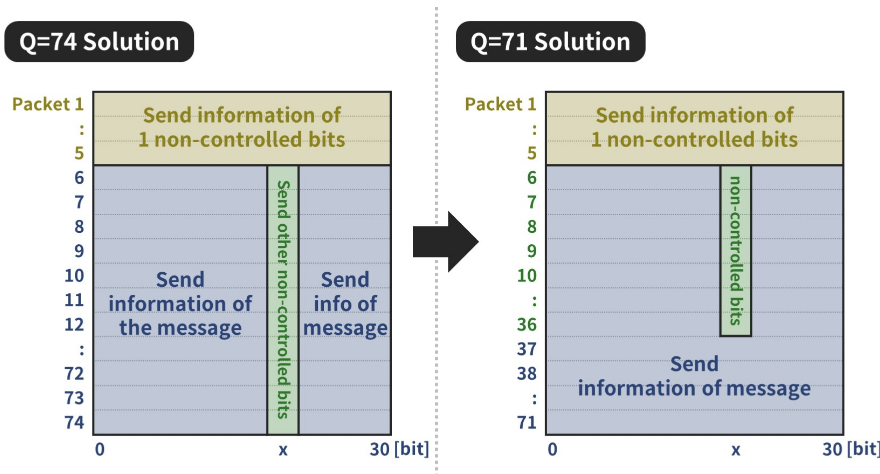
However, with the idea of "sending only 1 non-controlled bit first", we can achieve $Q = 74$ by the following algorithm.

1. For the first 5 packets, Aisha sends the index of 1 non-controlled bit x, with the idea of Subtask 1.
2. For the next 69 packets, Aisha sends the index of remaining 15 non-controlled bits **in x-th bit in the packet**, and embed the content of message in the other non-controlled bits.

The idea of the algorithm is described in the figure below. Note that since we can send $69 \times 15 = 1035$ bits for the information of the message, which is greater than 1025 bits, the number of packets is enough.



We can make a small improvement: since 31 bits is enough to send the information of the other 15 non-controlled bits, we can send the information of M even in the x-th bit after $(31 + 5 =)$ 36-th packets (see figure below). With this idea, we can achieve $Q = 71$. It suffices to get around 80 points.



5. Subtask 2: $Q = 66$ (100 points)

The idea for the full solution is completely different from Chapter 4. First, let's think about sending all information of non-controlled bits in 5 packets.

The idea is to send "the index of the next non-controlled bit". Let $a_1, a_2, \ldots, a_{16}$ be the index of non-controlled bits in increasing order (that is, $a_i < a_{i+1}$). If Aisha sends the binary representation of a_{i+1} in the a_i-th bit of the packet (the exception is $i = 16$, in this case "a_{17}" is the alias of a_1), Basma can know the values of $a_1, a_2, \ldots, a_{16}$ regardless of Cleopatra's action.

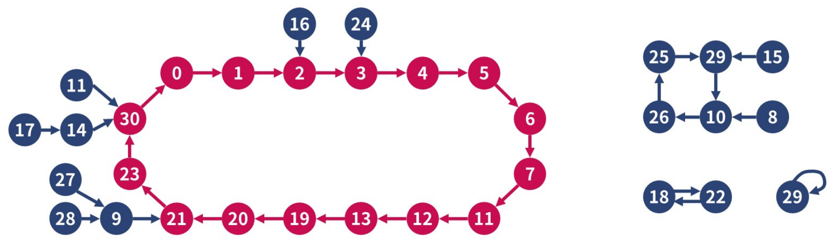
For example, if $a = [0, 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 19, 20, 21, 23, 30]$, the first 5 packets should be like the following figure. In the ($a_1 = 0$)-th bit, since the binary representation of $a_2 = 1$ is 00001, Aisha should send 1, 0, 0, 0, 0 in the 1, 2, 3, 4, 5-th packets, respectively. Note that for the controlled bits like the 8-th bit, anything is OK because Cleopatra can change freely.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Non-controlled	●	●	●	●	●	●	●	●				●	●	●						●	●	●		●							●
Packet 1	1	0	1	0	1	0	1	1	(anything is ok)	(anything is ok)	(anything is ok)	0	1	1	(anything is ok)	(anything is ok)	(anything is ok)	(anything is ok)	(anything is ok)	0	1	1	(anything is ok)	0	(anything is ok)	(anything is ok)	(anything is ok)	(anything is ok)	(anything is ok)	(anything is ok)	0
Packet 2	0	1	1	0	0	1	1	1				0	0	1						0	0	1		1							0
Packet 3	0	0	0	1	1	1	1	0				1	1	0						1	1	1		1							0
Packet 4	0	0	0	0	0	0	0	1				1	1	0						0	0	0		1							0
Packet 5	0	0	0	0	0	0	0	0				0	0	1						1	1	1		1							0

So, how does Basma know the values of $a_1, a_2, \ldots, a_{16}$? Let the binary representation of b_i be the i-th bit along the first 5 packets that Basma received (e.g. in the figure above, $b_0 = 1$), and consider about the following directed graph.

- Vertex: 31 vertices labeled from 0 to 30
- Edge: For each i ($0 \leq i \leq 30$), there is an edge $i \to b_i$

In this graph, regardless of Cleopatra's action, there is **exactly one** "cycle of length 16", and this cycle contains vertex $a_1, a_2, \ldots, a_{16}$. Therefore, by finding a cycle of length 16, Basma can know all non-controlled positions. An example of a graph is depicted below (red vertices correspond to non-controlled bits).



Therefore, we can achieve $Q = 70$, because another 65 packets are needed to send the message.

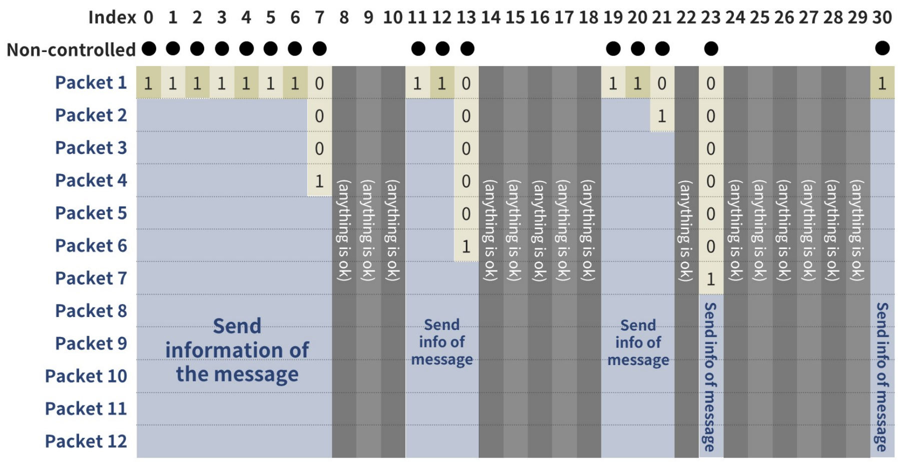
Moreover, $Q = 69$ can be achieved with a small improvement: instead of sending the binary representation of a_i, think about sending $(a_{i+1} - a_i) \bmod 31$. Since we can easily prove that $1 \leq [(a_{i+1} - a_i) \bmod 31] \leq 16$, only 4 packets are needed to send the information of non-controlled bits.

The final step towards the full solution is to change the encoding method: from binary encoding to one-hot encoding. More precisely, we can encode the value as follows.

Value	1	2	3	4	5	...
Encoded	`1`	`01`	`001`	`0001`	`00001`	

The content of packets when $a = [0, 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 19, 20, 21, 23, 30]$ is described in the following figure. Note that the remaining non-controlled bits, that are not used in encoding of $a_{i+1} - a_i$ (blue area of the figure), will be used to send the information of S.

Since one-hot encoding requires $\sum[(a_{i+1} - a_i) \bmod 31] = 31$ bits, to get 100 points ($Q = 66$), $66 \times 16 - 31 = 1025$ bits are remaining. And, it is exactly the length of message to send. A magical solution!


Index 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Non-controlled
Packet 1
Packet 2
Packet 3
Packet 4
Packet 5
Packet 6
Packet 7
Packet 8
Packet 9
Packet 10
Packet 11
Packet 12
1 1 1 1 1 1 1 0
0
0
1
1 1 0
0
0
0
0
1
1 1 0
1
0
0
0
0
0
0
1
1
(anything is ok)
Send information of the message
Send info of message
Send info of message
Send info of message
Send info of message