

Interactive, output only & Communication Task



Alex Poon 28/3/2015

★ Batch Task

- Most of the problems you have encountered are batch task
- We can read all input data before process in batch task
- But there is some non-batch task
- For example, interactive task, output only task, communication task

✦ Interactive Task

- Your program will interact with a judging program
- More precisely, your program will ask problems(queries)
- The judging program will answer your queries
- You need to solve the problems by the answer of queries
- Usually, there are constraints limiting how many times you can query

★ Example - Comparing Game

- HKOJ M1431
- There are N distinct numbers $A[i]$, $1 \leq i \leq N$
- You can ask the judging program : is $A[i]$ greater than $A[j]$?
- The judging program will answer : Yes or No
- You need to find i, j where $A[i], A[j]$ is the maximum & minimum among all N number
- You can query at most $1.5N$ times

★ Example - Comparing Game

- Note that in the problem, the only information giving to you initially is N (How many number are there)
- You must get the correct answer by asking the judging program for some information
- This is Interactive Task

✦ Interactive Task

- The problem statement will guide you the format of interacting
- You must follow the format when you are querying
- Or else you recieve 0 marks

★ Example - Comparing Game

- For example, the format of this problem is :
- Interact through standard I/O, which mean you query by output your query in specific format, and read the answer of the query

★ Interactive Task

- You should output “Q x y” for query, denote is $A[x]$ greater than $A[y]$.
- The judging program output “1” if the answer is Yes, otherwise 0
- You should output “A a b” for answering, denote $A[a]$ is the maximum value, $A[b]$ is the minimum value

★ Example - Comparing Game

- ***Note that
- Before you are reading the answer of queries, You need to flush the buffer first
- flush the buffer : that mean asking the judging program to output the data(answer) at this moment
- flush is a command in pascal/C++
- C++: `fflush(stdout);` Pascal: `flush(output);`

★ Sample Partial Solution - Pascal

```
readln(n);
For i := 1 to n do begin
  largest := True;
  For j := 1 to n do
    if i <> j then begin
      writeln('Q ', i, ' ', j);
      flush(output);
      readln(ans);
      if ans = 0 then largest := 0;
    end;
  if largest = 1 then writeln('A ', i);
end;
```

**We simplify the problem to find the maximum only

```
// Output your query
```

```
// flush the buffer
```

```
// read the answer given by the judging
program, note that the answer is 0/1
```

```
// Output your answer
```

★ Sample Partial Solution - C++

```
scanf("%d", &n);
for(int i = 1; i <= n; i++) {
    bool largest = 1;
    for(int j = 1; j <= n; j++)
        if (i != j) {
            printf("Q %d %d\n", i, j);
            fflush(stdout);
            scanf("%d", &ans);
            if (ans == 0) largest = 0;
        }
    if (largest == 1) printf("A %d\n", i);
}
```

**We simplify the problem to find the maximum only

// Output your query

// flush the buffer

// read the answer given by the judging program, note that the answer is 0/1

// Output your answer

✦ Interactive Task

- Other than interaction through standard I/O (HKOI uses)
- Another type of interaction : through library (IOI/NOI use)
- The library includes some functions for you to call for query & answer
- Usually, the problem statement will state how to use the functions for querying & answering

✦ Interactive Task

<http://www.ioinformatics.org/locations/ioi13/contest/day2/cave/cave.pdf>

★ Tips for Interactive Task

- Usually an Ad-hoc problem, there is no specific algorithm to solve interactive task
- You need observations to solve the task
- Don't be afraid when encounter this type of problem
- Regard the constraints, how many query you can ask
- The constraint may give you tip of the solution
- e.g. $N = 1000$, make 10 query at most
- The solution may be about Binary Search

★ Tips for Interactive Task

- Do some tasks before TFT, IOI, NOI as it often appears in these contest. e.g. TFT 11, 12, 13, IOI 13, 14...
- Practise Problems : HKOJ M1431, T054, T113, T134, 01084

★ Output Only Task

- All input data is open
- You only need to submit the output file of your program
- i.e. No time limit, memory limit
- Usually, there are no fast & optimal solution
- But require a near-optimal solution
- The marks you get base on the accuracy of your program
- There is usually a formula for calculating your score

★ Example - Disparity

- JOI Open Contest 2013
- <http://cms.ioi-jp.org/data/open-2013/disparity-en.pdf>
- Optimal Solution - Exhaustion (But too slow, run > 5 hrs)
- Require Solution - Greedy, Branch & Bound, Heuristic (Near-optimal solution, Fast solution)

★ Heuristic Search

- For greedy, branch & bound, learn it in the corresponding powerpoint in prepare by HKOI
- Heuristic search: For each searching phrase, define a function for estimating is current phrase can reach a near-optimal solution. If Yes, continou to search, else stop

★ Tips for Output Only Task

- Again, there are no specific algorithm for solving them
- Observe all the input data first, there maybe simple solution that can handle special case (e.g. case 2 in disparity)
- Handle different case by different program
- Observe the scoring formula, to know how to balance the accuracy & the time
- Do small case by Hand !! (see case 1, 2 in disparity)

★ Tips for Output Only Task

- Again, don't be afraid of this type of problem, although it is difficult /.\
- Observe first, then code
- Be creative

★ Communication Task

- You need to write 2 programs (or 2 modes)
- Inputs of the 2 programs are different
- Usually, program A will get more information, then you need to encrypt the useful information to a 01 string and send to program B
- Program B uses the encrypted 01 string to solve the problems
- Score depends on the length of the 01 string you send

★ Communication Task

- Flow - Run program A \rightarrow Input of program B = the output of program A \rightarrow Run program for solving the problem
- Ad-hoc Problems. No specific algorithm to solve them =[
- Observation!!!

★ Example - Lost Sequence (TFT2014)

- There are a sequence of numbers. Length = $2N$
- Program A : Read the first N elements of the sequence
- Program B : Read the Last N elements of the sequence
- Problem : Find the Longest increasing Subsequence of the original $2N$ sequence, there can be at most 0.1% error
- Communication : Send a 01 string from program A to B for getting information about the first N elements

★ Example - Lost Sequence (TFT2014)

- In other word, what you need to do is :
- Program A : Output a 01 string to program B
- Program B: Output the length of the LIS of the $2N$ sequence

★ Example - Lost Sequence (TFT2014)

- Trivial Solution
- Program A :
 - Use binary system to represent the first N elements
 - Send to program B directly
- Program B:
 - Decrype the 01 string into decimal number
 - Run a LIS algorithm

★ Communication Task

- Important technique, decimal \rightarrow binary (bases converting)
- Learn it in data processing/string processing =]

★ Optimization Method

- How to get higher marks ? Uses fewer bits
- Usually, there are 2 ways to compress your 01 string
 1. Reduce the number of data you need to send
 2. Reduce the number of bit you need in order to send 1 datum
- For 1, there is no general method, you need observations so that you may find that you can solve it by sending fewer data
- For 2, there are some frequent-used method

★ Optimization Method 1

- E.g. you need to convert N numbers range from 1 - 10 to binary
- 1, 10, 11 ... 1001, 1010
- Normally, you can use 4 bits to encrype 1 number as :
- $2^3 = 8$ not enough for 10, $2^4 = 16$ enough =]
- Then you need $4N$ bits to encrype all numbers

★ Optimization Method 1

- A better way is to Combine the Bits
- Use 10 bits to encrype 3 numbers
- $2^{10} = 1024 > 1000$ enough =]
- So, you need only $10/3 * N = 3.33N$ bits

★ Optimization Method 2

- You need to encrype N numbers range tend to $[0, 2^{20} - 1]$ this time
- Normally, you use 20 bits for encrypting 1 number, totally 20N bits
- However : how do you encrype 1 ? → 00000 00000 00000 00001
- How do you encrype 5 ? → 00000 00000 00000 00101
- The leading zeros are annoying =[

★ Optimization Method 2

- Reduce the unneeded Leading zeros
- e.g. if the number < 1024 , use 10 bits only, else use 20 bits
- then how to encrype 1 \rightarrow 00000 00001 \rightarrow fewer bits =]
- Wait, when I am decrypting (in Program B), how do I know I use 10/20bits to encrype this numbers ?

★ Optimization Method 2

- Add a signal
- e.g. If the first bit is 0, then the following 10 bits are in 1 group
- If the first bit is 1, then the following 20 bits are in 1 group
- How to encrype 1 → 0 00000 00001
- How to encrype 1000000 → 1 11110 10000 10010 00000
- The main point is to where is the cut off ... < 1024 ?? < 4096 ??
- And how many cut off you should add

★ Optimization Method 3

- Huffman Coding
- A method that “zip” use for zipping
- Note that we simply change a number to binary for encrypting
- Can we use other 01 string to represent a number ? Yes actually
- e.g. We change 10 → 01 instead of 1010, change 1 → 1010 instead of 01
- If we represent frequent-appear numbers by a shorter 01 string → better

- That what Huffman Coding try to do
- <http://www.csie.ntnu.edu.tw/~u91029/Compression.html>

★ Optimization Method 3

- Strength of Huffman coding → can apply to majority of problems & cases
- Drawback → Difficult to code
- So, use huffman code if both method 1, 2 cannot be applied

★ Example - Lost Sequence (TFT2014)

- Back to problem
- How can we get higher marks ?
- Tips
- Learn how to find LIS by binary search
- Note that you can have 0.1% error
- The number is from $[0, 2^{20}-1]$ → reduce leading zero
- make the sequence have more small number (think about partial sum)

★ Tips for Communication Task

- Observation + Creative !!
- Observe the scoring formula → how many bits you can use in order to full
- Bases changing
- Optimization method list above

- Again, don't be afraid !!!!
- Practise : T144, S141

★ Conclusion

- Actually, they are common type of problems
- Frequent appear in TFT, IOI, NOI
- Try to do some problems before TFT
- Don't be afraid
- From my opinion, these problem are relatively easy to get partial score but difficult to full
- Be Creative Be Creative Be Creative !!!