Interactive, output only & Communication Task

Alex Poon 09/04/2016

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
- Given N, there are N unknown integers A[1] … A[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

Comparing Game

SAMPLE RUN

Input	Output	Explanation
3		$n = 3$
	Q 1 2	Is card 1 larger than card 2?
0		No. Card 2 is larger.
	Q 3 1	Is card 3 larger than card 1?
1		Yes.
	Q 2 3	Is card 2 larger than card 3?
1		Yes.
	A 2 1	Max card: 2, Min card: 1.

The problem guide you how to interact with the judge program

In this problem, you should output "Q x y" when make the following query is card x larger then card y

You should flush the buffer before reading the respond of the query

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 : Summary

- *****Interactive task is NOT saying the task is difficult
- Interactive task maybe very standard or adhoc, it depends
- Remember to flush the buffer

- erm...just like normal-batch task, nothing special actually
- Often appear in TFT (11, 12, 13), IOI(13, 14, 15)

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 - Maze

- IOI 2010 Day Q3

- Given a N * M grip, some of them a occupied
- Construct a path in the grip s.t.
- it touch the boundary of the grip exactly 1 time

http://ioi2010.org/Tasks/PlayAlong/Maze/tthh.shtml

Example - Maze

- 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

- Usually an adhoc task
- Easy to get partial score but very difficult to full
- Small case can be done by hand and type your output by keyboard on your own
- Read the input data carefully in order to get partial score

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 encrype the useful information to a 01 string and send to program B

- Program B uses the encryped 01 string to solve the problems
- Score depends on the length of the 01 string you send

Communication Task

- Flow of communication task
- Run program A
- The output of program A = Input of program B
- Run program B
- Grade your program by output of program A/B

Dividing the cities (HKOI 2014)

- Given N nodes and M edges and 10 colour
- Construct a way to colour the nodes s.t. no two nodes connected by an edge has same colour
- N <= 3000, M <= 10000

Dividing the cities

Program A:

- Read the node and edge and a valid way to colour the node
- Output a 01 string with length L

Program B:

- Read the node and edge and the 01 string
- Output a valid way to colour the node

Mark depends on L

Dividing the cities

Trivial Solution:

Program A: Encode the whole answer provide to an 01 string

Program B: Decode the 01 string back to the whole answer

e.g. The answer provide is 1 2 3 1 9

We may use 4 bits to encode 1 answer (2^4 = 16 > 10)

1 -> 0001, 2 -> 0010, 3 -> 0011, 9 -> 1001.

The 01 string become : 00010010001100011001

In practical, this solution use 4 * N bits and get 25 marks

Dividing the cities

- We should reduce length of the 01 string in order to get higher marks

- We have 2 direction to reduce L
  - Reduce the bits we need to encode the same number of data
  - Reduce the number of data we need to encode

Dividing the cities

- Optimization 1
- We use 4 bits (can represent 16 comb.) to encode 1 answer(10 comb.)
- It is waste
- Better to use 10 bits (can represent 1024 comb.)
  to encode 3 answer (1000 comb.)

e.g. Answer is 10 3 5 -> we first convert it to 0 base -> 9 2 4 -> 924

e.g. Use 10 bits to represent 924 by change it to binary

- Can get 42 marks

Dividing the cities

- Optimization 2
- Note that if a node has < 10 neighbours
- No matter what the colour of its neighbours is
- There must exist a colour s.t. no collision occur

Therefore, if a node has < 10 neighbours, we don’t need to pass its answer from prog A to prog B

Dividing the cities

- There are at most 10000 edges
- Which mean there are at most 10000 * 2 / 10 = 2000 nodes has >= 10 neighbour
- Combine 2 optimizations, we use only 2000 / 3 * 10 = 6667 bits
- We can get 70 marks

Communication Task

- Usually, we need some observation to reduce number of data we need to send

- But we have some standard way to reduce the bits we need to encode same amount of data

Method 1 - Reduce leading zero

- e.g. We need to encode A[1..N] where A[i] <= 2^20
- We need 20 bits to encode 1 data

Improvement

- e.g. if the number < 1024, use 10 bits only, else use 20 bits
- then how to encrype 1 → 00000 00001 → fewer bits =]

- Wait, when I am decryping (in Program B), how do I know I use 10/20bits to encrype this numbers ?

Method 1 - Reduce leading zero

- 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

Method 2 - Huffman Coding

- Main idea :
- Previously, we only use the binary representation of a number to encode it
- Actually, we can use only 01 string to encode a number
- e.g. we can use 101 to encode 1, 01 to encode 2, 00 to encode 3 etc...

- If the frequency of a data is high, we hope the encryption of it is short
- That’s the main idea of huffman coding

Method 2 - Huffman Coding

- e.g. to encode 1 2 3 3 3 3 3 3 3
- We can simply use 01 to encode 1, 10 to encode 2, 11 to encode 3
- We need 2 * 9 = 18 bits in total

OR

- We can use 10 to encode 1, 11 to encode 2, 0 to encode 3
- Note that {10, 11, 0} no one is the prefix of another
- We need 2 * 2 + 7 = 11 bits only

Summary: Communication Task

- Important technique, decimal <-> binary (bases converting)
- Communication task vary a lot, sometimes we need standard algorithm knowledge (such as knowledge about graph)
- Sometimes (should be Usually) we need adhoc observations
- Easy to get partial score but difficult to full

Practise

- The best way to perform better in non-batch task is to practise!!!

M1431, T054, T113, T124, T134, T144, S141, I1321, I1011, I1021, I1023, …...

1. 調皮的小孩：download.noi.cn/T/noi/noi2002A.pdf
2. Towns：http://olympiads.kz/ioi2015/day2/towns-en.pdf
3. Parrots：http://www.ioi2011.or.th/hsc/tasks/EN/parrots.pdf
4. Rail:

http://www.ioinformatics.org/locations/ioi14/contest/day1/rail/rail.pdf