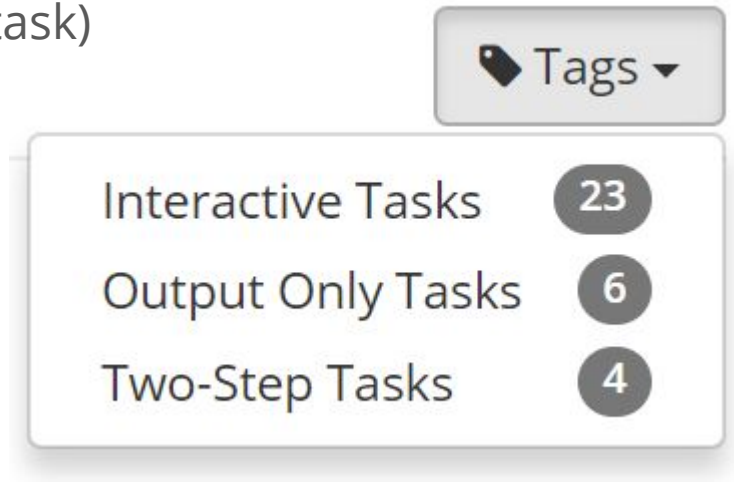
Interactive, Output-only & Communication tasks

Percy Wong {percywtc}

Tasks Categorization

- Batch task
- Interactive task
- Output-only task
- Communication task (a.k.a. Two-steps task)



How Important?

- [IOI2010] Cluedo
- [IOI2010] Languages
- [IOI2010] Memory
- [IOI2013] Cave
- [IOI2014] Game
- [IOI2015] Scales
- [IOI2016] Unscrambling a Messy Bug
- [IOI2017] The Big Prize
- [IOI2010] Maze
- [IOI2012] Pebbling Odometer
- [IOI2017] Nowruz
- [IOI2011] Parrots

- [TFT2011] Stones Rearrangement
- [TFT2012] Debug!
- [TFT2013] The Forgotten Triangle
- [TFT2016] Model Answer II
- [TFT2017] Constellation
- [TFT2014] Lost Sequence

- **Interactive**
- **Output-only**
- **Two-steps**

What's the difficulty?

- Unfamiliar style
- You may not be able to understand these problems during the contests, if you are the first time facing new types of tasks

- Feedback from inexperienced contestants after TFTs
  - 「唔知條題目講乜」
  - 「睇唔明題目」

Interactive task

- Your program will interact with the judging program
- You can consider it as: (suitable for most interactive tasks)
  - Your program asks some questions
  - The judging program answers your questions
  - Repeat the aboves until you can solve “something”
  - (Just like playing MASTERMIND / Guess the Number)
- Usually, there will be limits on number of questions asked
- Or, your score is determined by questions asked

M1431 Comparing Game

- **N** distinct cards not revealed to you
- Your goal: find where are the maximum and the minimum cards

- Question you may ask:
  - "Is card **X** larger than card **Y**?"

- Ask no more than ⌊1.5**N**⌋ questions

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.

M1431 Comparing Game

- How can our program ask question?
  - using standard I/O

Pascal version

```
writeln('Q ', x, ' ', y);
flush(output);   // IMPORTANT
readln(result);
```

C/C++ version

```
printf("Q %d %d\n", x, y);
fflush(stdout);  // IMPORTANT
scanf("%d", &result);
```

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.

M1431 Comparing Game

Pascal version (sample partial solution)

```
for i := 1 to N do
  for j := 1 to N do
  begin
    counter := 0;
    if (i <> j) then
    begin
      writeln('Q ', i, ' ', j);
      flush(output);
      readln(result);
      if (result = 1)
        counter := counter + 1;
    end;
    if (counter = N - 1) then
      bigIndex := i;
    if (counter = 0) then
      smallIndex := i;
  end;
```

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.

M1431 Comparing Game

C/C++ version (sample partial solution)

```
for (int i = 1; i <= N; i++)
  for (int j = 1; j <= N; j++) {
    counter = 0;
    if (i != j) {
      printf("Q %d %d\n", i, j);
      fflush(stdout);
      scanf("%d", &result);
      if (result == 1)
        counter++;
    }
    if (counter == N - 1)
      bigIndex = i;
    if (counter == 0)
      smallIndex = i;
  }
```

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.

Interactive task

- The example just now performs interaction through standard I/O
  - writeln / printf
  - readln / scanf
- Some interactive tasks are using another way
  - through the grader program
  - you will be given a template code
  - you will ask questions / get feedback by calling some given functions

I0501 Divisor Game

- An unknown integer **K** within the range [1, **N**]
- Your goal: find the value of **K**

- Question you may ask:
  - "Is the number **K** divisible by some integer **x**?"

- Ask minimal questions

Assume that the grader calls your function `play(1000)`.

Call	Returns	Explanation
`isDivisibleBy(10)`	`1`	K is divisible by 10.
`isDivisibleBy(100)`	`1`	K is divisible by 100.
`isDivisibleBy(1000)`	`0`	K is not divisible by 1000.
`isDivisibleBy(200)`	`0`	K is not divisible by 200.
`isDivisibleBy(300)`	`0`	K is not divisible by 300.
`isDivisibleBy(500)`	`0`	K is not divisible by 500.
`isDivisibleBy(700)`	`0`	K is not divisible by 700.

Your function `play` should return 100, the number K Alice has in mind.

I0501 Divisor Game

- What is given?

TEMPLATE

Download official grader files. Please note that you may need to make

Pascal | C/C++

```pascal
unit submission;

interface
function isDivisibleBy(M: longint): longint; cdecl; external;

var
  // TODO: global variables can be declared here

implementation
function play(N: longint): longint; cdecl; export;
var
  // TODO: implementation
begin
  // TODO: implementation
end;
end.

```

TEMPLATE

Download official grader files. Please note that you may

Pascal | C/C++

```cpp
#ifdef __cplusplus
extern "C" {
#endif
int isDivisibleBy(int M);
int play(int N);
#ifdef __cplusplus
}
#endif

// TODO: global variables can be declared here

int play(int N) {
  // TODO: implementation
}

```

I0501 Divisor Game

- How can our program ask question?
  - using grader functions

Pascal version

```
result := isDivisibleBy(x);
```

C/C++ version

```
result = isDivisibleBy(x);
```

Assume that the grader calls your function `play(1000)`.

Call	Returns	Explanation
`isDivisibleBy(10)`	`1`	K is divisible by 10.
`isDivisibleBy(100)`	`1`	K is divisible by 100.
`isDivisibleBy(1000)`	`0`	K is not divisible by 1000.
`isDivisibleBy(200)`	`0`	K is not divisible by 200.
`isDivisibleBy(300)`	`0`	K is not divisible by 300.
`isDivisibleBy(500)`	`0`	K is not divisible by 500.
`isDivisibleBy(700)`	`0`	K is not divisible by 700.

Your function `play` should return 100, the number K Alice has in mind.

I0501 Divisor Game

- You cannot compile the program even if you have completed `play()`
  - it's because the main program is missing
- You cannot successfully test the program
  - it's because the function `isDivisibleBy()` is not implemented
  - this function is implemented by the judging program
  - you are only required to implement `play()`

- So what can we do to test our program?

I0501 Divisor Game

- So what can we do to test our program?
  - we can implement the remaining functions to test our part
  - and remember to delete these lines before submit

```
int n,ans,ct;

int isDivisibleBy(int M) {
//  printf("isDivisibleBy(%d)\n",M);
    ct++;
    return ans % M == 0;
}

int main () {

    while (true) {
        ct = 0;
        n = rand() % 1000000 + 1;
        ans = rand() % n + 1;

//      scanf("%d%d",&n,&ans);

        int ret = play(n);
        printf("count = %d\n",ct);
        if (ret != ans) {
            printf("Error : return %d [n = %d, ans = %d]\n",ret,n,ans);
            break;
        }
        printf("OKOK  : return %d [n = %d, ans = %d]\n",ret,n,ans);
    }

    return 0;
}
```

- Being familiar with problems using grader

  IS VERY DIFFICULT !!!

- Please make use of the Online Judge,
  and try these tasks

Output-only task

- Formal Definition
  - Input files are given to you
  - You are not required to upload any source codes, just the output files

- Actual meaning
  - No need to worry about failing some unknown cases, all cases are revealed :D
  - No time limits / memory limits (actually there are... TL = 5hrs, ML = your machine)
  - You can even solve the cases manually :D :D :D

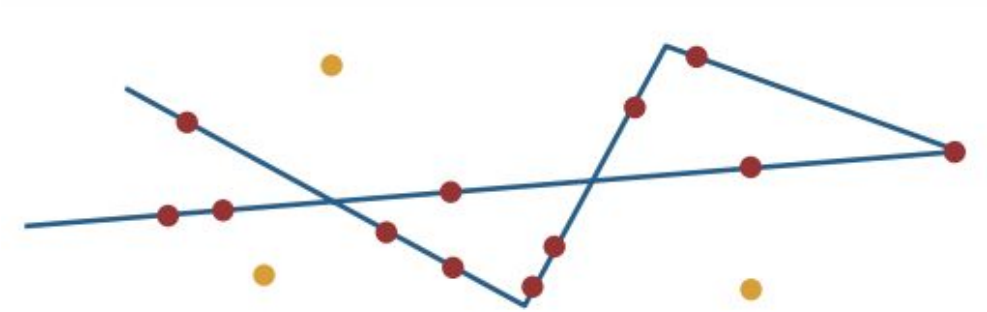
Output-only task

- Common stuffs?
  - not expecting optimal solutions (or not even exist)
  - some formulas to determine how good your outputs are (and how much you score)
  - good-enough solutions can get good-enough scores

- What you can do?
  - Usually there exists some small cases (can be manually solved)
  - You can write programs to analyze the cases / solve the cases
  - You can even solve cases separately with different approach and codes

T174 Constellation

- Given a set of **N** points with integral coordinates lying on xy-plane
- Build a polyline consisting of **V** points, connecting most points



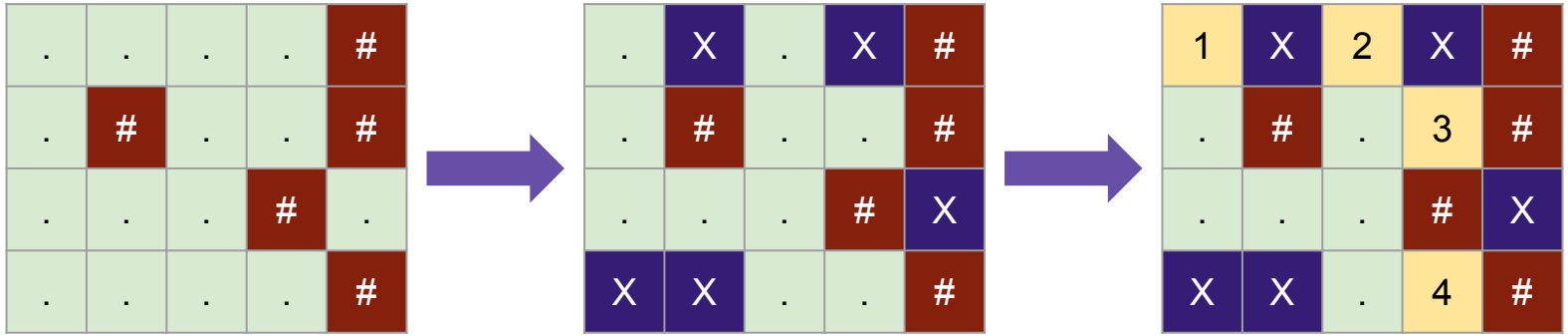
T174 Constellation

- 10 cases in total (each 10pts)
- Scoring are based on number of points you connect
  - $10 \times 10^{(P - V) / (T - V)}$

Case	Input	Output	N	V	T
1	stars1.txt	const1.txt	25	2	10
2	stars2.txt	const2.txt	49	13	49
3	stars3.txt	const3.txt	12	7	12
4	stars4.txt	const4.txt	80	3	18
5	stars5.txt	const5.txt	200	41	180
6	stars6.txt	const6.txt	90	20	90
7	stars7.txt	const7.txt	40	11	28
8	stars8.txt	const8.txt	120	25	115
9	stars9.txt	const9.txt	200	35	185
10	stars10.txt	const10.txt	200	50	200

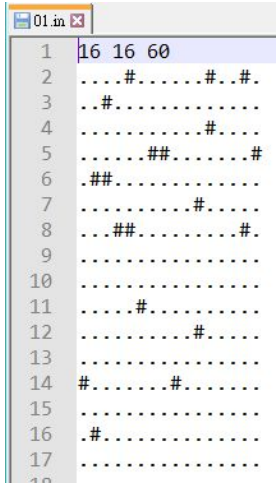
I1711 Nowruz

- Given an **n** × **m** grid with some obstacle cells
- Build a maze that has as many「堀頭路」(dead end) as possible
  - 「堀頭路」(dead end): cell that has exactly one free neighbour



I1711 Nowruz

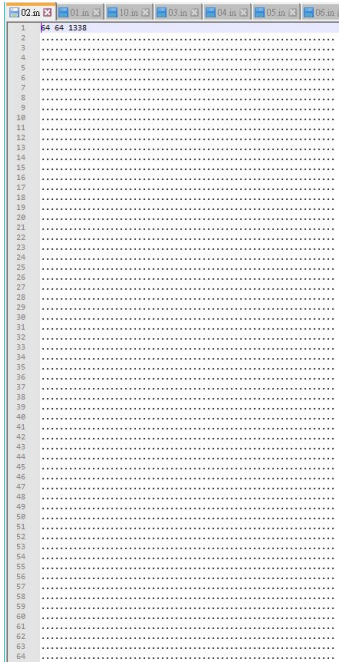
- Here are some of the input cases (there are total 10 cases)



01.in

16 × 16

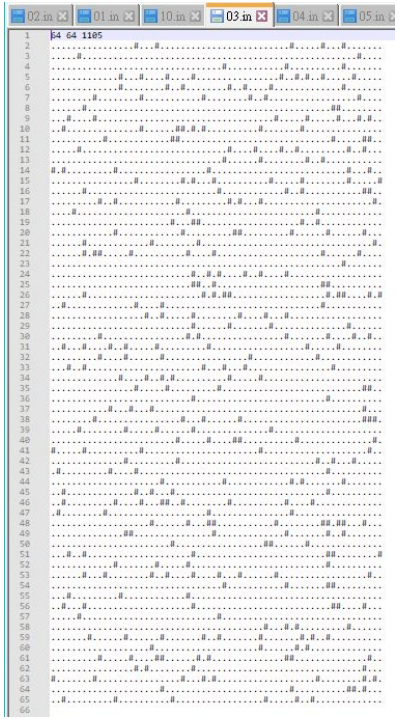
random



02.in

64 × 64

empty



03.in

64 × 64

random

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

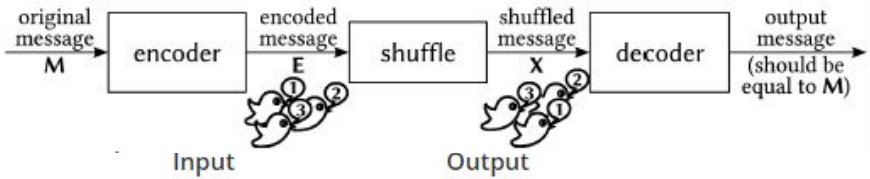
Communication task (Two-steps task)

- You have to write two subprograms (or two modes)
- Judging flow:
  - [source input] → [program mode A] → [output A]
  - [input based on output A] → [program mode B] → [final output]
- Your score usually depends on the length / efficiency of [output A]

- Program mode A
  - build up a strategy that can transfer more information with shorter length
- Program mode B
  - build up a strategy to interpret the [output A] and extract some useful data

I1123 Parrots

- Original message **M** consists of at most 64 integers within [0, 255]



original message M

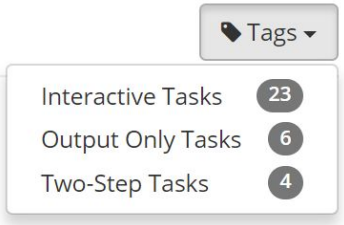
shuffle

your final output

	Input	Output
1	E 1 3 10 30 20 Mode E	7 7 3 2 70 15 20 3
2	D 1 3 7 3 20 70 15 2 3 7 Mode D	10 30 20

Conclusion

- Just like constructive task, non-batch task is another type of problems
  - NOT LIMITED by any algorithms, topics
  - therefore, no standard rules to deal with them
  - again, “practice makes perfect”
  - as long as you solve / take a look at more non-batch tasks,
  - more techniques / experiences you can accumulate
- From the history of Team Formation Test,
  - non-batch tasks often appear :)
  - good luck :)



Practice Tasks

- [IOI2010] Cluedo
- [IOI2010] Languages
- [IOI2010] Memory
- [IOI2013] Cave
- [IOI2014] Game
- [IOI2015] Scales
- [IOI2016] Unscrambling a Messy Bug
- [IOI2017] The Big Prize
- [IOI2010] Maze
- [IOI2012] Pebbling Odometer
- [IOI2017] Nowruz
- [IOI2011] Parrots

- [TFT2011] Stones Rearrangement
- [TFT2012] Debug!
- [TFT2013] The Forgotten Triangle
- [TFT2016] Model Answer II
- [TFT2017] Constellation
- [TFT2014] Lost Sequence

• **Interactive** • **Output-only** • **Two-steps**