

7. Optimizing Mo's Algorithm (moo)

1 sec / 3 sec

60 points

Juku is preparing for the Especially Important Olympiad and discovered the following problem:

Sums in an Array

You are given an array A with length N and Q queries of the form (l, r) .

For each query give the result $A[l] + A[l + 1] + \dots + A[r]$ as answer.

It is known that $N = Q = 10^5$.

Juku wrote the following pseudocode to solve the problem:

```
lptr = 1, rptr = 1, sum = A[1], ops = 0
```

```
for each query (ql, qr):
```

```
  while rptr < qr:
```

```
    rptr += 1
```

```
    sum += A[rptr]
```

```
    ops += 1
```

```
  while lptr > ql:
```

```
    lptr -= 1
```

```
    sum += A[lptr]
```

```
    ops += 1
```

```
  while rptr > qr:
```

```
    sum -= A[rptr]
```

```
    rptr -= 1
```

```
    ops += 1
```

```
  while lptr < ql:
```

```
    sum -= A[lptr]
```

```
    lptr += 1
```

```
    ops += 1
```

```
  print sum
```

The running time of this code exceeded the time limit. However, Juku vaguely remembered that in some lecture someone said something about some “Mo’s algorithm” and started thinking: “Could I reorder the queries such that the program would fit within time limit? Moreover, how fast could I make my program by just reordering queries?”

You’re given Q queries of the form (l, r) . Additionally, it is guaranteed that the queries are uniformly distributed (see the remark). Reorder queries such that by running the above pseudocode the final value of variable `ops` would be as small as possible.

Grading. In this problem points are given according to the “goodness” of the solution. More precisely, let’s assume that Juku’s code is run on your ordering. The final value of the variable `ops` is used to evaluate the “goodness” of your solution. Every test gives a maximum of 6 points, there are 10 tests in total. The ordering produced by your solution gets

$$\min\left(6, 6 \cdot \frac{2 \cdot 10^7}{\text{ops}}\right) \text{ points for a test.}$$

Input. The first line of the input file `moosis.txt` contains two space-separated integers N and Q ($N = Q = 10^5$). On the following Q input lines there are two space-separated integers l ja r on each ($1 \leq l \leq r \leq N$).

Output. The output file `mooval.txt` should contain Q lines. Each line should have two space-separated integers l and r . The queries in the output file must be a permutation of the queries in the input file.

Example.	<code>moosis.txt</code>	<code>mooval.txt</code>
	7 6	2 7
	1 5	1 2
	2 7	3 6
	4 4	1 2
	1 2	4 4
	3 6	1 5
	1 2	

Note that the input in this example is **not** a valid input file, as $N = Q = 10^5$ does not hold. It only exists to demonstrate the format of input and output files.

Remark. The actual input files have been generated using the following code:

```
print(100000, 100000)
for i in 1...100000:
    l = random(1, 100000)
    r = random(1, 100000)
    if (l < r):
        print(l, r)
    else:
        print(r, l)
```

Here, `random(1, 100000)` returns an integer x such that $1 \leq x \leq 100\,000$ and all possible return values are equally probable.