

Programming Assignment #2 (due on-line 1pm, November 17, 2019)

Submission URL & Online Resources:

<https://cool.ntu.edu.tw/courses/469/assignments/3850> (Prof. James Chien-Mo Li's class)

<https://cool.ntu.edu.tw/courses/527/assignments/3851> (Prof. Iris Hui-Ru Jiang's class)

<https://cool.ntu.edu.tw/courses/543/assignments/3820> (Prof. Yao-Wen Chang's class)

Problem: Maximum Planar Subset

Given is a set C of n chords of a circle (see Figure 1 (a)). We assume that no two chords of C share an endpoint. Number the endpoints of these chords from 0 to $2n - 1$, clockwise around the circle (see Figure 1 (c)). Let $M(i, j)$, $i \leq j$, denote the number of chords in the maximum *planar subset* (i.e., no two chords overlap each other in the subset) in the region formed by the chord \overline{ij} and the arc between the endpoints i and j (see Figure 1 (d)). As the example shown in Figure 1 (a), $M(2, 7) = 1$, $M(3, 3) = 0$, and $M(0, 11) = 3$. You are asked to write a program that computes the number of chords in the maximum planar subset in a circle of n chords, i.e., compute $M(0, 2n - 1)$, and reports the details of each chords, as shown in Figure 1 (b).

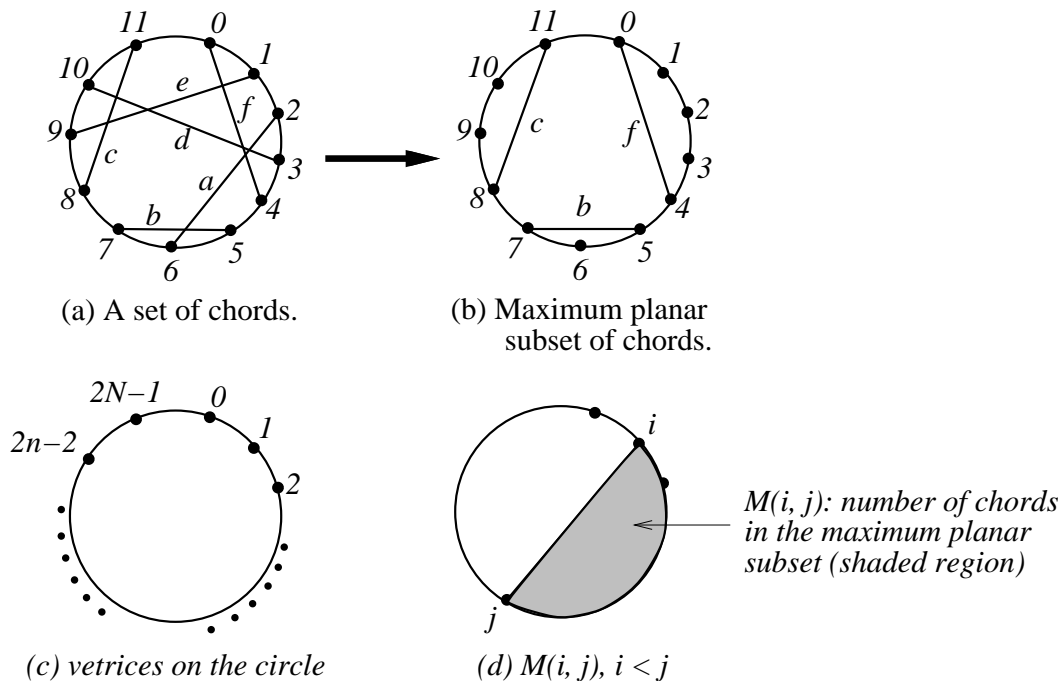


Figure 1: Maximum planar subset.

Input

The input consists of an integer $2n$, $1 \leq n \leq 20,000$, denoting the number of vertices on a circle, followed by n lines, each containing two integers a and b ($0 \leq a, b \leq 2n - 1$), denoting two endpoints of a chord. A single “0” (zero) in the input line signifies the end of input.

Output

The output file reports the number of chords in the maximum planar subset in the input circle of n chords, followed by a list of the two endpoints for each resulting chord in the maximum planar subset (sorted by the first endpoint in the increasing order).

Here is an input/output example (see Figure 1):

Sample Input	Sample Output
12	3
0 4	0 4
1 9	5 7
2 6	8 11
3 10	
5 7	
8 11	
0	

可能不只一種解，但規則要符合，數量要對

Command-line Parameter:

The executable binary must be named as “mps” and use the following command format.

```
./mps <input_file_name> <output_file_name>
```

For example, if you would like to run your binary for the input file `12.in` and generate a solution named `12.out`, the command is as follows:

```
./mps 12.in 12.out
```

Required Files:

You need to create a directory named `<student_id>_pa2/` (e.g. `b06901000_pa2/`) which must contain the following materials:

- A directory named `src/` containing your source codes (e.g. `maxPlanarSubset.cpp`): only `*.h`, `*.hpp`, `*.c`, `*.cpp` are allowed in `src/`, and no directories are allowed in `src/`;
- An executable binary named `mps`;
- A makefile named `makefile` or `Makefile` that produces an executable binary from your source codes by simply typing “make”: the binary should be generated under the directory `<student_id>_pa2/`;
- A text readme file named `readme.txt` describing how to compile and run your program.

Then please use the following command to compress your directory into a `.tgz` file:

```
tar zcvf <filename>.tgz <your directory>
```

The submission file should be named as `<student_id>_pa2.tgz` (e.g. `b06901000_pa2.tgz`). For example, if your student ID is `b06901000`, then you should use the command below.

```
tar zcvf b06901000_pa2.tgz b06901000_pa2/
```

Please submit your *.tgz* file to the NTU COOL system before **1pm, November 17, 2019 (Sunday)**. To make sure that your submission satisfies all the requirements, we provide a script `checkSubmitPA2.sh` to check your *.tgz* file by the command below:

```
bash checkSubmitPA2.sh <your submission>
```

For example,

```
bash checkSubmitPA2.sh b06901000_pa2.tgz
```

Language/Platform:

1. Language: C or C++.
2. Platform: Linux.

Evaluation:

An individual score per test case is determined by the correctness of the output result as well as the file format. The runtime limit for each case is 60 seconds. For fair evaluation, please apply the **-O3** optimization for the compilation. Three input test cases are provided and more hidden test cases will be used for the final test.

For any questions, please email Chen-Hao Hsu at r07943107@ntu.edu.tw, Yu-Jie Cai at r07943111@ntu.edu.tw, or Chien-Hao Tsou at d08943011@ntu.edu.tw. Thank you so much for your cooperation. Have fun with this programming assignment!