

National Taiwan University of Science and Technology
Department of Electrical Engineering
Algorithm Design and Application, Spring 2016
Programming Assignment #1
Maximum Planner Subset (due May 7, 2016 (Saturday) on-line)

1. Problem Description

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 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).

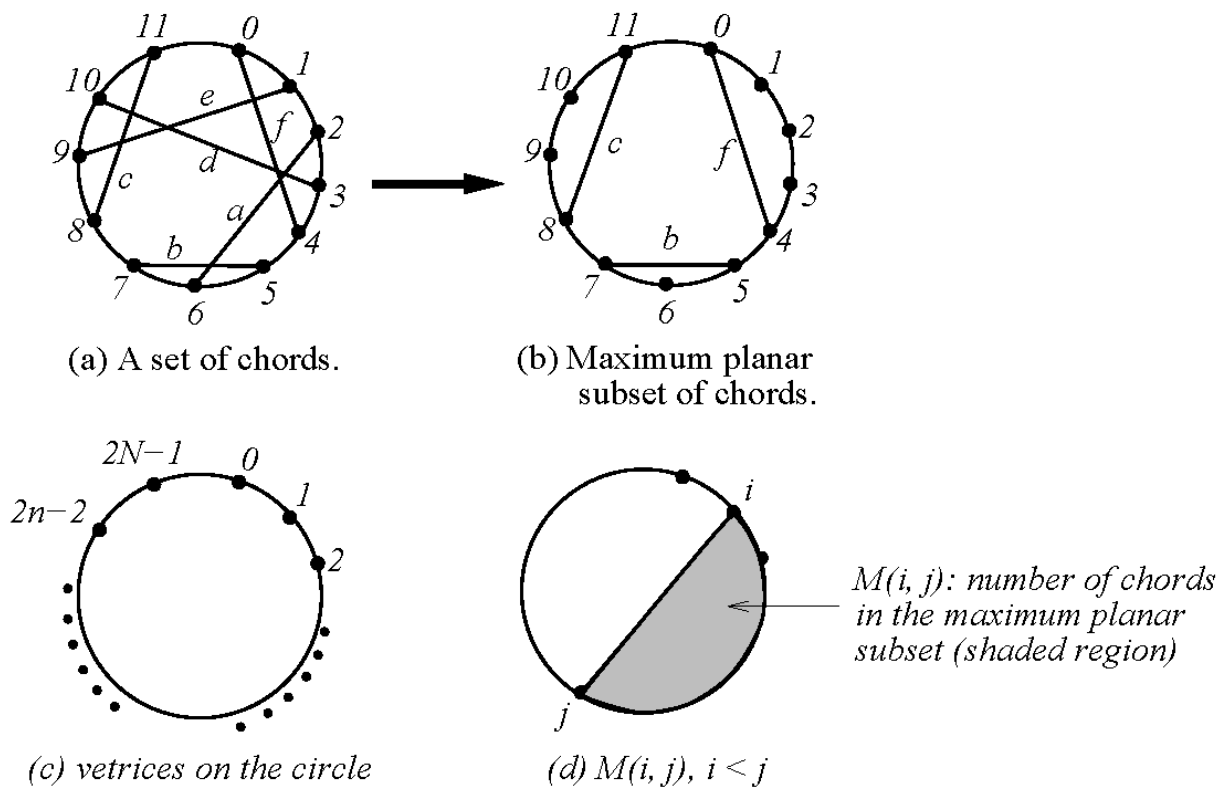


Figure 1: Maximum planner subset.

2. Input

The input consists of an integer $2n$, $1 \leq n \leq 10000$, 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.

3. 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 of 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	

4. Language/Platform

- (a) Language: C or C++.
- (b) Platform: Unix/Linux or Windows.

5. Command-line Parameter

In order to test your program, you are asked to add the following command-line parameters to your program (e.g., MPS.exe 12.in 12.out):

[executable file name] [input file name] [output file name]

6. Submission

You need to submit the following materials in a .tar or a .zip file (e.g., m10407400-p1.zip) at the course website by the deadline: (1) source codes, (2) executable binaries, and (3) a text readme file (readme.txt) stating how to build and use your programs.

7. Grading Policy

This programming assignment will be graded based on (1) the correctness, (2) readme.txt, (3) solution quality, and (4) running time. Please check these items before your submission.

8. Online Resources

Sample input files (*.in), sample parser, and sample readme.txt can be found at the course website.