National Taiwan University of Science and Technology
Department of Electrical Engineering

Algorithm Design and Application, Fall 2023

Programming Assignment #1

Maximum Planner Subset (due November 12, 2023 (Sunday) 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 $M(i,j),\ i \le j$, 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).

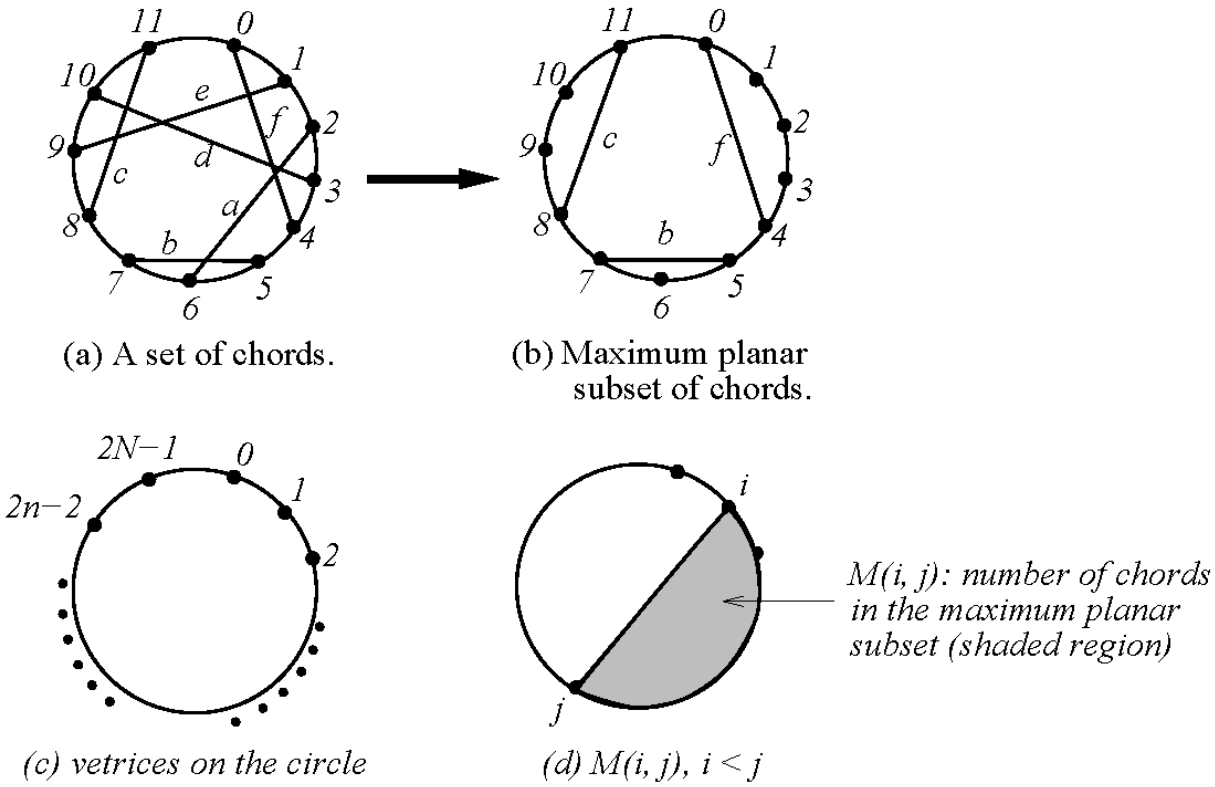

Figure 1: Maximum planner subset.

2. Input

The input consists of an integer $2n,\ 1 \le n \le 10000$, denoting the number of vertices on a circle, followed by *n* lines, each containing two integers *a* and *b* ($0 \le a, b \le 2n-1$), 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 0 4 1 9 2 6 3 10 5 7 8 11 0	3 0 4 5 7 8 11

4. Language/Platform

(a) Language: C or C++.

(b) Platform: Unix/Linux. A tutorial for installing virtual Linux system on PC is available on Moodle.

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 12.in 12.out):

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

**Penalty will be given if your program fails to use command-line parameters!

6. Submission

You need to submit the following materials in a compressed [student id]-p1.tgz file (e.g., b11007000-p1.tgz) at the course website by the deadline: (1) source codes, (2) Makefile, and (3) a text readme file (readme.txt) stating how to build and conduct your program.

Hints:

- The compressed file [student id]-p1.tgz file contains only a single folder named [student id]-p1 (e.g., b11007000-p1). Use only lowercase letters for the compressed file and folder names.
- Only a compressed file in the *.tgz format will be accepted.
- Do not submit files or folders other than those specified above.
- Please ensure that your work can be successfully executed in the Linux environment.

**If the above requirements are not met, penalties will be imposed

7. Grading Policy

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

8. Online Resources

Sample input files (*.in) and a sample submission file (including a sample parser, a sample Makefile, and a sample readme.txt) can be found on the course website.