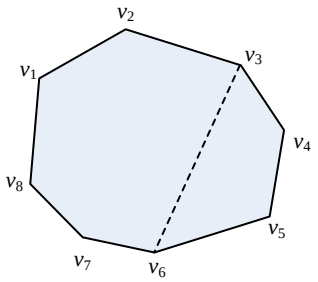
IT 427, Design and Analysis of Algorithms

Programming Assignment 7: Polygon Triangulation

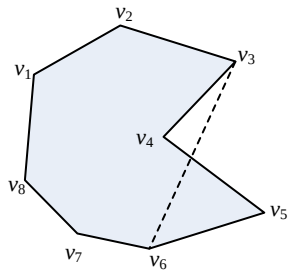
Due date: November 4, 2024, Monday, 11:59 PM 40 points (30 on programs, 10 on report)

For this assignment, you will use dynamic programming technique to find the optimal triangulation of convex polygons. The problem is defined as follows. Let $\langle v_1, v_2, \ldots, v_n \rangle$ with $3 \leq n \in \mathbf{N}$ represent an n-sided polygon P where $v_1, \ldots, v_n$ are the vertices and line segments $\overline{v_1v_2}, \overline{v_2v_3}, \ldots, \overline{v_iv_{i+1}}, \ldots, \overline{v_{n-1}v_n}$, and $\overline{v_nv_1}$ are the n sides of the polygon.

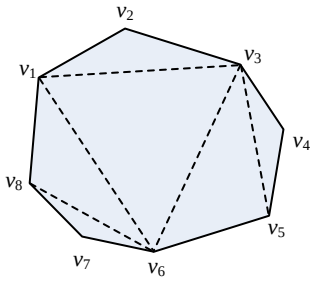
We say that two vertices v_i and v_j are non-adjacent if line segment $\overline{v_iv_j}$ is not a side of the polygon, and in this case, we say that $\overline{v_iv_j}$ is a *chord* of the polygon. A polygon is *convex* if no points on any chords fall outside the region of the polygon. For example, polygon A is convex, but polygon B is not because of chord $\overline{v_3v_6}$ of B. Given a convex polygon, a *triangulation* is a set of chords such that no two chords intersect in the interior (i.e., if two chords intersect, the intersection must be one of the vertices) and separate the polygon into triangles.



Polygon A



Polygon B



Triangulation of A

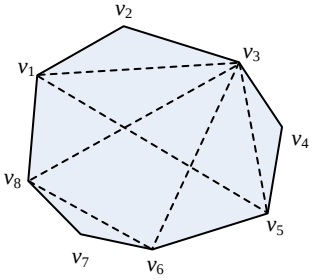


Figure 1

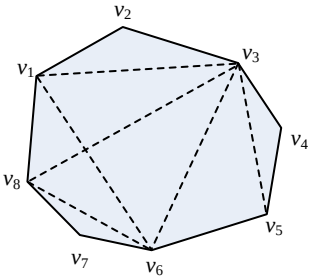


Figure 2

For example, polygon A is triangulated by $T = \{\overline{v_6v_8}, \overline{v_6v_1}, \overline{v_6v_3}, \overline{v_1v_3}, \overline{v_3v_5}\}$ into 6 triangles $\triangle v_1v_2v_3$, $\triangle v_1v_3v_6$, $\triangle v_1v_6v_8$, $\triangle v_3v_4v_5$, $\triangle v_3v_5v_6$, and $\triangle v_8v_6v_7$, while the sets of chords in Figure 1 and Figure 2 are not *triangulations*. We will assign a weight to each triangle in a triangulation as we consider such a weight as the cost of forming a triangle. Let $w(\triangle v_iv_jv_k)$ be the length of the perimeter of the triangle, i.e, $\|\overline{v_iv_j}\| + \|\overline{v_jv_k}\| + \|\overline{v_kv_i}\|$ in Euclidean distance. We aim to find an optimal triangulation with the minimum cost. Note that, a side shared by two triangles should not be counted twice. Thus, the weight of the triangulation of polygon A above is the length of A's perimeter plus $\|\overline{v_1v_3}\|+\|\overline{v_1v_6}\|+\|\overline{v_3v_5}\|+\|\overline{v_3v_6}\|+\|\overline{v_6v_8}\|$ where each of $\|\overline{v_1v_6}\|$ and $\|\overline{v_3v_6}\|$ is shared by two triangles but counted once.

- Make a directory `asg7` under your `IT427`, i.e., `~/IT427/asg7/`. All needed files for this assignment should be saved under your `~/IT427/asg7` before run `submit427.sh`.
- Check the contents of my `/home/ad.ilstu.edu/cli2/Public/IT427/asg7` and copy all files to your own `~/IT427/asg7`.

Program requirement: The name of the program should be `triangulate.py`. I will compile and run your program on our Linux server as follows, where `polygons0.txt` is the input file. If your program fails to compile, you will get 0 point. I may test your program on different files.

```
python3 triangulate.py polygons0.txt
```

Input: The input text file contains several polygons, where each polygons is saved as follows:

```
   --- x y ----
-623.0835569821  353.5607499179
-347.9882857123  145.6459683932
 108.7816253885   37.9197242061
-575.8528206080  434.5455901060
-601.1391590708  422.1545134071
```

Note that, the polygons in the file may not be convex. There are a few files in the public directory for your testing. I may use another files to test.

Output: Since we triangulate only convex polygons, the program need to test whether the polygon is convex first. If it is not convex, the program simply indicates the test result and skip triangulating. If it is convex, the program should find the optimal triangulation and print the weight and all chords formatted as follows.

```
......

**Polygon no.3 is convex.
Optimal Triangulation:  minimum weight = 2674.459555
3 chords:  (xi, yi)-(xj, yj)
(676.738982, 417.733849)-(128.094859, 462.573694)
(312.973766, 724.048676)-(128.094859, 462.573694)
(312.973766, 724.048676)-(111.215660, 492.126774)

**Polygon no.4 is not convex.

**Polygon no.5 is convex.
Optimal Triangulation:  minimum weight = 7948.610802
4 chords:  (xi, yi)-(xj, yj)
(-2122.140157, 564.397365)-(-2032.929560, 990.640723)
(-2122.140157, 564.397365)-(-1088.544949, 826.225002)
(86.315659, 318.795282)-(-1088.544949, 826.225002)
(86.315659, 318.795282)-(45.304661, 446.409512)

......

All by (your name here)
```

Submission: Programs (30 points) and Reports (10 points)

Submission details are same as the previous assignment. Run the submission script with the submission number changed to 6, but you can use the same secret name as follow:

```
bash /home/ad.ilstu.edu/cli2/Public/IT427/submit427.sh peekapoo 7
```

Note that, since I will keep updating `submit427.sh` for different assignment, you have to run the script from my `/home/ad.ilstu.edu/cli2/Public/IT427/` directly for the most recent updated version, i.e., don't copy it to your own directory.

1. Programs: 30 points. Submission on Linux server.

   The score is based on the correctness and the programming style, which includes efficiency, appropriateness of data structures, and documentation of your programs. At the beginning of every program file, put a section of comments including (1) your full name, (2) student ID, (3) a pledge of honesty that you do not copy/modify from other's codes and (4) a declaration of your copyright that no one else should copy/modify the codes. You will receive:

   (a) 95 $\sim$ %: No error with a good programming style.
   (b) 80 $\sim$ %: Minor error and fair programming style.
   (c) 60 $\sim$ %: Some error and not so good but acceptable programming style.
   (d) 40 $\sim$ %: Too many error and bad programming style, but meaningful.
   (e) 20 $\sim$ %: Compilable but not working and the program must show reasonable trying.
   (f) 0 $\sim$ %: : Fail to meet any of aforementioned qualities or plagiarism involved.

2. Report: 10 points. Submission through Canvas.

   You have to write up a report and prepare it in pdf format. You don't have to put program output on the report as I will run and exam your program directly on some different input files.

   The report should includes brief descriptions of your program, summary of the methods, data structures, and efficiency analysis on time and space in details in terms of big-O notations. If there is any difficulties encountered in this assignment, you can report it. If your analysis is not clearly related to your program with sufficient justification, your report score will not be higher than 50%.