Zehao Li
cs130b
Programming Assignment #3

RUN:

```
make
./prog3 < [input file]
(Then the result will be in output.txt)
```

PLOT:
Then when you need to change the path of the input file name in tri.m to plot the picture.

1, How does the principal of optimality apply?

In my program, I input all the vertices at first. I save them into two vector<Points> P1 and P0. All the points in P1 has the z of 1 and all the points in P0 has the z of 0.

Then, I start to implement my dynamic programming.

The method is mainly to construct a table, which has the P1 point label and P0 point label as two dimensions. Table has (number of points in P1 + 1) columns and (number of points in P0 +1) rows. Because I need to use all the points to get a closed surface.

The coordinate (i , j) in this table stands for the cost of the closed area which conclude the 0~i points in P1, and 0~j points in P0. The cost is the current surface area. The edge of two vertices in the table stands for a triangle. I could calculate the area of this triangle using my helper function.

To solve the problem that we have many situations of the start points pair, I use an int start to track the start point in P1. In this way, we could cover all the cases.

For each point in the table, I save an object Info to record the previous point value in the table and the cost at current point. Saving this value, I could easily backtrack all the triangles I have for the least surface area.

In the table, I start from (0,0) to calculate the cost. Each point of the table can only calculate the cost from left point and upper point. I use the recurrence function in 2nd question to calculate the cost. After completing the table, I could get the least area surface value from the last point in the table, (number points in P1, number points in P0). Then, I will compare it withe the global vector minrecord to update the minrecode value.

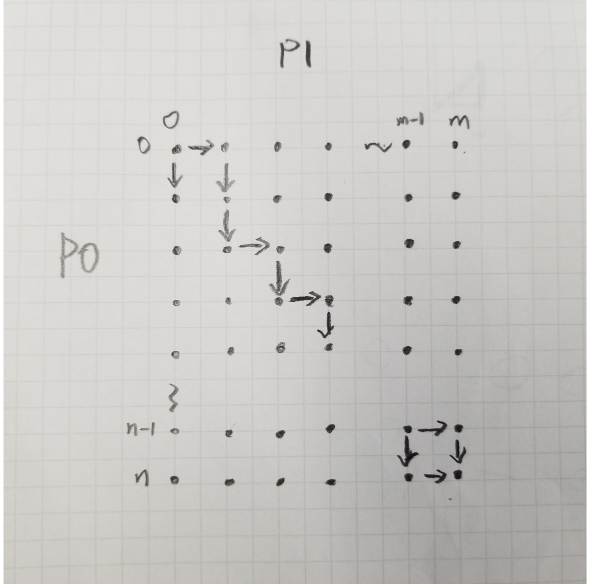
At last, I use print to print all the result and output to output.txt

2, What is the recurrence relation used in solving this problem?

The coordinate (i , j) in this table stands for the cost of the closed area which conclude the 0~i points in P1, and 0~j points in P0. The cost is the current surface area.

cost(i, j)=
0 ,if i==0 && j==0
min{ (cost(i,j-1)+area(P1[i],P0[j],P0[j-1])),(cost(i-1,j)+area(P1[i],P1[i-1],P0[j])) } ,else

3, How does the table of partial solutions look like? How to construct such a table?



We could see the table for 001_boxPoints.txt, start from P1[0] and P0[0]

P0 II P1	0	1	2	3	4(0)
0	(-1,-1),0	(0,0),0.5	(1,0),1.20711	(2,0),1.91421	(3,0),2.41421
1	(0,0),0.5	(0,1),1	(1,1),1.5	(2,1),2.20711	(3,1),2.91421
2	(0,1),1.20711	(1,1),1.5	(1,2),2	(2,2),2.5	(3,2),3.20711
3	(0,2),1.91421	(1,2),2.20711	(2,2),2.5	(2,3),3	(3,3),3.5
4(0)	(0,3),2.41421	(0,4),2.91421	(2,3),3.20711	(3,3),3.5	(3,4),4

I have declared how to construct the table in question1:

Construct a table, which has the P1 point label and P0 point label as two dimensions. Table has ([number of points in P1] + 1) columns and ([number of points in P0] +1) rows. Because I need to use all the points to get a closed surface.

For each point in the table, I save an object Info to record the previous point value in the table and the cost at current point. Saving this value, I could easily backtrack all the triangles I have for the least surface area.

To solve the problem that we have many situations of the start points pair, I use an int start to track the start point in P1. In this way, we could cover all the cases. That is to say we must construct (number of points in P1) tables.

I start from (0,0) to calculate the cost. Each point of the table can only calculate the cost from left point and upper point. I use the recurrence function in 2nd question to calculate the cost. After completing the table, I could get the least area surface value from the last point in the table, (number points in P1, number points in P0).