CS 4820, Spring 2014 **Homework 4, Problem 1**
Name: Piyush Maheshwari
NetID: pm489
Collaborators: None

(1) If P is a convex polygon in the plane, a *triangulation* of P is a way of dividing it up into triangles whose corners are vertices of P. More precisely, a triangulation of P consists of a set of line segments $\mathbf{L} = \{L_1, \ldots, L_r\}$ and a set of triangles $\mathbf{T} = \{T_1, \ldots, T_s\}$ satisfying the following properties.

1. For each segment $L_i \in \mathbf{L}$, the endpoints of L_i are vertices of P.
2. For each triangle $T_j \in \mathbf{T}$, the sides of T_j are segments in $\mathbf{L}$.
3. No two segments in $\mathbf{L}$ cross each other in the plane. In other words, any two segments $L_i, L_j \in \mathbf{L}$ are either disjoint or they have a unique point of intersection that is an endpoint of both segments.
4. Every side of the polygon P is a side of exactly one triangle $T_j \in \mathbf{T}$.
5. Every segment $L_i \in \mathbf{L}$ that is not a side of P is a side of exactly two triangles $T_j, T_k \in \mathbf{T}$.

Suppose that P is a convex polygon in the plane and that you are given an input consisting of the following information:

- a list of the vertices $v_1, v_2, \ldots, v_n$ of P, in clockwise order;
- a positive integer cost $c(i, j)$ that denotes the cost of creating a line segment from v_i to v_j. These costs are defined whenever $1 \leq i < j \leq n$.

For any triangulation of P, the cost of the triangulation is defined to be the sum of the costs of the line segments included in the set $\mathbf{L}$. Design an algorithm to compute the minimum cost of a triangulation of P.

Solution

Lemma 1. *Both vertices of any edge (u,v) of the polygon will be connected to some vertex w in any possible triangulation.*

Proof. Suppose in any possible triangulation vertices u,v of an edge are connected to different vertices x and y. Without loss of generality we can assume that u,v,x,y follow a clockwise order. If $x \neq y$ then the edge u,v is not part of any triangle since the line segments they are part of must not cross each other. This means there must be some other x' and y' between x and y which are also connected to vertices u,v. Now using the same argument on x' and y' we can find another pair x" and y". Eventually these vertices will converge since there are a finite number of vertices and hence u, v will be connected to the same vertex. This proves the lemma. □

Now let opt(i,j) be the minimum cost of triangulation of the polygon formed by vertices i to j taken in clockwise order. Now from lemma 1 we know that there must be some k between i and j to which i and j must be connected and they will form a triangle. So the cost of such a triangulation will be the cost of the triangulation of polygon i to k, plus the cost of triangulation of polygon k to j plus the cost of the triangle formed for i , k , j. Since there are only i - j -1 such values of k, we can calculate the minimum over all values of k and we will get the minimum cost of triangulation of the polygon between i and j.

opt(i , j) = c_{ij} for $\forall i, j$ such that j - 1 = 1

opt(i , j) = $min_{k=i+1toj-1}$ opt(i ,k) + opt(k , j) + c_{ij}

This recurrence gives us a way to calculate opt(i,j) for all values of i, j such that 1 $\leq$ i $<$ j $\leq$ n . The final answer will be given by opt(1,n). We can observe that while calculating opt(i,j) we depend upon sub problems opt(m,n) such that $|m - n| < |i - j|$. Thus we can calculate all the value of opt(i,j) in increasing order of $|i - j|$.

Algorithm 1

```
procedure Min-Triangulation
    opt[i, j] ← INF ∀ i, j such that 1 ≤ i < j ≤ n        ▷ Initialize the table opt
    for all len = 1 to n do
        for all i = 1 to n - len - 1 do
            j ← i+len
            if j = i + 1 then
                opt(i,j) ← c_ij                              ▷ This is the base case
            else
                for all k = i+1 to j-1 do
                    opt(i,j) ← min (opt(i,j), opt(i ,k) + opt(k , j) + c_ij
    return opt(1,n)
```

Proof of correctness

We can prove this by induction. We claim that opt(i,j) contains the minimum triangulation cost of the polygon formed by vertices from i to j.

Base case : When j - 1 = 1, this means that the polygon is just a edge of the polygon and the cost of triangulating it would be simply the cost of that edge.

opt(i , j) = c_{ij} for $\forall i, j$ such that j - 1 = 1

Inductive Step : From lemma 1 we know that the polygon formed by vertices from i to j can be triangulated by choosing a k between i and j, triangulating the polygon from i to k , then triangulation the polygon from k to j and then adding the triangle i , k , j. So the minimum cost of triangulating a polygon which has a triangle i, k, j can be found by adding the min cost of triangulating polygon (i,k) plus min cost of triangulating polygon (k,j) plus the cost of triangle i, k, j. And to find the min cost of triangulation of polygon i,j we can consider all such k's and take a minimum over them. Note that we are adding only cost of c_{ij} since the cost of the other vertices of the triangle will be included in the other two vertices.

optimal(i , j) = $min_{k=i+1toj-1}$ optimal(i ,k) + optimal(k , j) + c_{ij}

From our inductive hypothesis we know that opt(i,k) and opt(k, j) are the optimal costs of the triangulating the polygons i,k and k,j respectively. Hence we have

optimal(i , j) = $min_{k=i+1toj-1}$ opt(i ,k) + opt(k , j) + c_{ij} = opt(i,j)

Hence the induction holds.

Running Time

We have O(n^2) sub problems and each takes O(n) time (line 9-10) to solve. So the total running time is O(n^3)