

7. Give an efficient algorithm for the following problem. The input is an n sided convex polygon. Assume that the polygon is specified by the Cartesian coordinates of its vertices. The output should be the triangulation of the polygon into $n - 2$ triangles that minimizes the sums of the perimeters of the into triangles. Note that this is equivalent to minimizing the length of the cuts required to create the triangle.

Given an input of an array of points in clockwise order that form a convex polygon.

Recursive algorithm:

$minPerim(p_1, p_2, \dots, p_n) :$

base case: if $n=3$ return $\|p_1 + p_2\| + \|p_2 + p_3\| + \|p_1 + p_3\|$

otherwise: return

$$\min_{1 \leq x \leq n} (\underbrace{\|p_{x-1} + p_x\| + \|p_x + p_{x+1}\| + \|p_{x-1} + p_{x+1}\|}_{\text{Perimeter of triangle defined by } x} + \underbrace{minPerim(I \setminus p_x)}_{\text{Remaining polygon}})$$

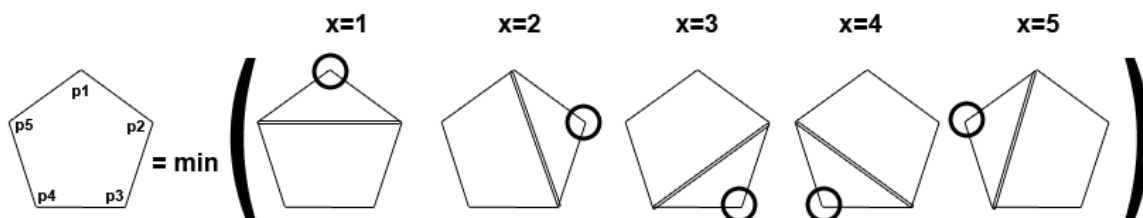


Figure 1: Illustration of the recursive call

The following is a conversion of the recursive solution into a bottom-up iterative solution for input I .

1. Calculate Base Cases first:

For $T = [i_1, i_2, i_3]$ in $\binom{n}{3}$, $i_1 < i_2 < i_3$: (that is, for every triangle permutation in I)
 $Lookup[T] = \|i_1 + i_2\| + \|i_2 + i_3\| + \|i_1 + i_3\|$. (the perimeter of the triangle)

2. Iterate!

For $m = 4$ to n : (Start with 4-gons, then 5-gons, \dots , to n -gons)

For $M = [i_1, i_2, i_3, \dots, i_m]$ in $\binom{n}{m}$: (For every m -gon)

$hash(M) =$

$$\min_{\substack{1 \leq x \leq m \\ \text{See Figure 1}}} (\underbrace{\|i_{x-1} + i_x\| + \|i_x + i_{x+1}\| + \|i_{x-1} + i_{x+1}\|}_{\text{Perimeter of triangle defined by } x} + \underbrace{Lookup(M \setminus i_x)}_{\text{Remaining } m-1\text{-gon}}) \quad \text{modulo } m+1$$

3. Find solution.

The solution to the n -gon is at $Lookup[I]$.