Lindsey Bieda and Joe Frambach
Dynamic Programming Problems
9.22.2011

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

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

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

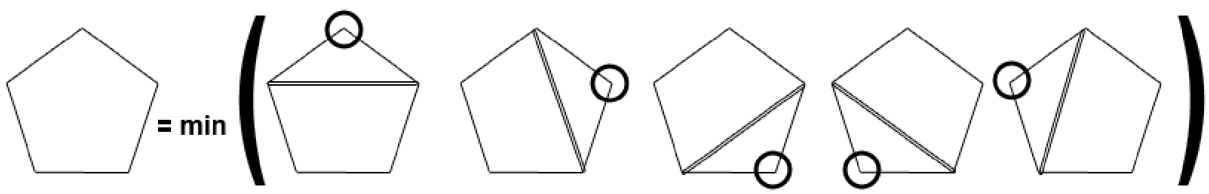

Figure 1: Illustration of the recursive call

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

1. Calculate Sbase 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, ..., to n-gons)
   For $M = [i_1, i_2, i_3, \ldots, i_m]$ in $\binom{n}{m}$: (For every m-gon)
   $hash(M) =$

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

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