Tutorial 2

Shivam Kumar Jha (17CS30033)

27-07-2019

1 Problem Statement

$P[1..n]$ is an input list of n points on the xy-plane. Assume that all n points have distinct x-coordinates and distinct y-coordinates. Let p_L and p_R denote the leftmost and the rightmost pointsof P, respectively. The task is to find the polygon Q with P as its vertex such that the following conditions are satisfied.

i) The upper vertex chain of Q is x-monotone (increasing) from p_L to p_R.

ii) The lower vertex chain of Q is x-monotone (decreasing) from p_R to p_L.

iii) Perimeter of Q is minimum.

You have to answer the following. Provide necessary figures/diagrams for explanations.

1. Develop the recurrences needed for DP, with clear arguments.
2. Design the algorithm and write its main steps.
3. Derive the time and space complexities of your algorithm.

2 Recurrences

Let $f(n, n)$ represent a x-monotone upper chain from p_1 to p_n and a x-monotone lower chain from p_n to p_1. Now, an intermediate of the same would be $f(i, j)$. It is easily observable that swapping i and j doesn't change the value of the result and hence both are equivalent $f(i, j) = f(j, i)$. This makes us interested only in the domain $1 \le i \le j \le n$.

$$f(i,j) = \begin{cases} 0, & i = j = 1 \\ f(j,i), & j < i \\ f(i,j-1) + d(j-1,j), & i < j-1 \\ \min f(i,k) + d(k,j), & i = j \text{ or } j-1 \end{cases} \tag{1}$$

Now, we look at the following four cases of the summarized recursion function above.

Case 1. $i == j == 1$: In this case the value of the path is zero since points coincide.
Case 2. $i > j$: In this case we shall take the value of $f(j, i)$ since we would have already crossed that point and we know that swapping i and j doesn't change the result.
Case 3. $i < j - 1$: In this case, we will take the path joining p_i to p_1 and then from p_1 to p_{j-1} along the line joining p_{j-1} to p_j. This is done since the point p_{j-1} lies on the upper vertex chain.
Case 4. $i = j | i = j - 1$: In this the optimal path is decided by traversing from i to j and finding the minimum possible value of $f(i, j-1) + d(j-1, j)$ for k between i to j.

3 Algorithm

4 Demonstration

5 Time and space complexities