

Algorithm TRIANGULATEMONOTONEPOLYGON(\mathcal{P})

Input. A strictly y -monotone polygon \mathcal{P} stored in a doubly-connected edge list \mathcal{D} .

Output. A triangulation of \mathcal{P} stored in the doubly-connected edge list \mathcal{D} .

1. Merge the vertices on the left chain and the vertices on the right chain of \mathcal{P} into one sequence, sorted on decreasing y -coordinate. If two vertices have the same y -coordinate, then the leftmost one comes first. Let u_1, \dots, u_n denote the sorted sequence.
2. Initialize an empty stack \mathcal{S} , and push u_1 and u_2 onto it.
3. **for** $j \leftarrow 3$ **to** $n - 1$
4. **do if** u_j and the vertex on top of \mathcal{S} are on different chains
5. **then** Pop all vertices from \mathcal{S} .
6. Insert into \mathcal{D} a diagonal from u_j to each popped vertex, except the last one.
7. Push u_{j-1} and u_j onto \mathcal{S} .
8. **else** Pop one vertex from \mathcal{S} .
9. Pop the other vertices from \mathcal{S} as long as the diagonals from u_j to them are inside \mathcal{P} . Insert these diagonals into \mathcal{D} . Push the last vertex that has been popped back onto \mathcal{S} .
10. Push u_j onto \mathcal{S} .
11. Add diagonals from u_n to all stack vertices except the first and the last one.