### Week 39: Advanced Geometry – Half-Plane Intersection & Rotating Calipers

**Topics:** - Half-Plane Intersection (HPI) - Convex Polygon Intersection - Rotating Calipers Method - Applications: Smallest Enclosing Rectangle, Diameter of Convex Polygon - Area and Perimeter Calculations of Intersections - Geometry Optimization Problems

**Weekly Tips:** - Half-plane intersection can be solved with line sorting and deque-based algorithm. - Rotating calipers can find farthest points, polygon diameter, and minimum bounding rectangle. - Convex polygon problems often reduce to geometry + rotating calipers. - Be careful with floating-point precision, use long double when necessary. - Always normalize line directions when working with HPI.

**Problem 1: Convex Polygon Diameter (Rotating Calipers)** **Link:** [CSES Polygon Diameter](https://cses.fi/problemset/task/2191/) **Difficulty:** Advanced

**C++ Solution with Explanation Comments:**

#include <bits/stdc++.h>  
using namespace std;  
struct Point{ long double x,y; };  
long double dist2(Point a,Point b){ return (a.x-b.x)\*(a.x-b.x)+(a.y-b.y)\*(a.y-b.y); }  
int main(){  
 int n; cin>>n; vector<Point> p(n);  
 for(int i=0;i<n;i++) cin>>p[i].x>>p[i].y;  
 // Convex polygon assumed (if not, run convex hull first)  
 long double ans=0; int j=1;  
 for(int i=0;i<n;i++){  
 while(dist2(p[i],p[(j+1)%n])>dist2(p[i],p[j])) j=(j+1)%n;  
 ans=max(ans,dist2(p[i],p[j]));  
 }  
 cout<<fixed<<setprecision(6)<<sqrt(ans)<<endl;  
}

**Explanation Comments:** - Rotating calipers checks antipodal points on convex polygon. - Maintains two pointers to find max distance efficiently. - O(n) after convex hull computation.

**Problem 2: Half-Plane Intersection (HPI)** **Conceptual Overview:** - Represent half-planes as lines + side. - Sort lines by angle. - Use deque to maintain feasible intersection polygon. - Final intersection polygon is convex.

**Applications:** - Polygon clipping. - Visibility regions. - Linear constraints optimization.

**Skeleton Code:**

// Pseudocode structure for Half-Plane Intersection  
struct Line{ Point p; Point dir; double ang; };  
  
// 1. Sort lines by angle  
// 2. Use deque to store intersection polygon edges  
// 3. Remove lines that make intersection infeasible  
// 4. Compute final polygon from deque

**Applications of Rotating Calipers:** - Smallest enclosing rectangle of convex polygon. - Width and thickness of polygon. - Farthest pair of points in convex hull.

**End of Week 39** - Master rotating calipers for polygon problems. - Learn Half-Plane Intersection for advanced geometry optimization. - Practice convex polygon problems and intersection calculations.