|  |
| --- |
|  |
| struct Point |
|  | { |
|  | int x, y; |
|  |  |
|  | Point operator -(Point p) |
|  | { |
|  | return {x-p.x, y-p.y}; |
|  | } |
|  |  |
|  | int dist() |
|  | { |
|  | return x\*x + y\*y; |
|  | } |
|  | }; |
|  |  |
|  | bool by\_x(Point &a, Point &b) |
|  | { |
|  | return a.x < b.x; |
|  | } |
|  |  |
|  | bool by\_y(Point &a, Point &b){ return a.y < b.y; |
|  |
|  |  |
|  | } |
|  |  |
|  | int n, ans=1e18; |
|  | int a[N], pref[N]; |
|  | Point pt[N]; |
|  |  |
|  | int solve(int L, int R){ |
|  | if(L==R) return 1e18; |
|  | int M=(L+R)/2; |
|  | sort(pt+L, pt+R+1, by\_x); |
|  | int d=min(solve(L, M), solve(M+1, R)); |
|  | int midx=pt[L+(R-L+1)/2].x; |
|  | vector<Point> v; |
|  | for(int i=L;i<=R;i++) |
|  | { |
|  | if(Point{pt[i].x-midx, 0}.dist()<d) |
|  | { v.push\_back(pt[i]); |
|  |  |
|  | } |
|  | } |
|  | sort(v.begin(), v.end(), by\_y); |
|  | for(int i=0;i<v.size();i++) |
|  | { |
|  | for(int j=i+1;j<v.size();j++) |
|  | { if(Point{0, v[i].y-v[j].y}.dist()>d) |
|  |  |
|  | break; |
|  | d=min(d, (v[i]-v[j]).dist()); |
|  | } |
|  | } |
|  | return d; |
|  |  |