Group Activity - 02

Faculty -: Dr. Saravanan D

group members -:
Abhishek Srivastava (19BCE10071)
Pravir Kadian (19BCE10006)

CODE -:

```
int quad(pair<int, int> p)
{
if (p.first >= 0 && p.second >= 0)
return 1;
if (p.first <= 0 && p.second >= 0)
return 2;
if (p.first <= 0 && p.second <= 0)
return 3;
return 4;
}
int orientation(pair<int, int> a, pair<int, int> b,
pair<int, int> c)
{
int res = (b.second-a.second)*(c.first-b.first) -
(c.second-b.second)*(b.first-a.first);
if (res == 0)
return 0;
if (res > 0)
return 1;
return -1;
}
bool compare(pair<int, int> p1, pair<int, int> q1)
{
pair<int, int> p = make_pair(p1.first - mid.first,
p1.second - mid.second);
pair<int, int> q = make_pair(q1.first - mid.first,
q1.second - mid.second);

int one = quad(p);
int two = quad(q);
if (one != two)
return (one < two);
return (p.second*q.first < q.second*p.first);
}
```

```
vector<pair<int, int>> merger(vector<pair<int, int> > a,

vector<pair<int, int> > b)

{
int n1 = a.size(), n2 = b.size();

int ia = 0, ib = 0;
for (int i=1; i<n1; i++)
if (a[i].first > a[ia].first)
ia = i;
for (int i=1; i<n2; i++)
if (b[i].first < b[ib].first)
ib=i;
int inda = ia, indb = ib;
bool done = 0;
while (!done)
{
done = 1;
while (orientation(b[indb], a[inda], a[(inda+1)%n1]) >=0)
inda = (inda + 1) % n1;
while (orientation(a[inda], b[indb], b[(n2+indb-1)%n2]) <=0)
{
indb = (n2+indb-1)%n2;
done = 0;
}
}
int uppera = inda, upperb = indb;
inda = ia, indb=ib;
done = 0;
int g = 0;
while (!done)
{
done = 1;
while (orientation(a[inda], b[indb], b[(indb+1)%n2])>=0)
indb=(indb+1)%n2;
while (orientation(b[indb], a[inda], a[(n1+inda-1)%n1])<=0)
{
inda=(n1+inda-1)%n1;
done=0;
}
}
int lowera = inda, lowerb = indb;
vector<pair<int, int>> ret;
```

```
int ind = uppera;
ret.push_back(a[uppera]);
while (ind != lowera)
{
ind = (ind+1)%n1;
ret.push_back(a[ind]);
}
ind = lowerb;
ret.push_back(b[lowerb]);
while (ind != upperb)
{
ind = (ind+1)%n2;
ret.push_back(b[ind]);
}
return ret;

}
vector<pair<int, int>> bruteHull(vector<pair<int, int>> a)
{
set<pair<int, int> >s;
for (int i=0; i<a.size(); i++)
{
for (int j=i+1; j<a.size(); j++)
{
int x1 = a[i].first, x2 = a[j].first;
int y1 = a[i].second, y2 = a[j].second;
int a1 = y1-y2;
int b1 = x2-x1;
int c1 = x1*y2-y1*x2;
int pos = 0, neg = 0;
for (int k=0; k<a.size(); k++)
{
if (a1*a[k].first+b1*a[k].second+c1 <= 0)
neg++;
if (a1*a[k].first+b1*a[k].second+c1 >= 0)
pos++;
}
if (pos == a.size() || neg == a.size())
{
s.insert(a[i]);
s.insert(a[j]);
} } }
vector<pair<int, int>>ret;
for (auto e:s)
```

```
ret.push_back(e);
mid = {0, 0};
int n = ret.size();
for (int i=0; i<n; i++)
{
mid.first += ret[i].first;
mid.second += ret[i].second;
ret[i].first *= n;
ret[i].second *= n;
}
sort(ret.begin(), ret.end(), compare);
for (int i=0; i<n; i++)
ret[i] = make_pair(ret[i].first/n, ret[i].second/n);
return ret;
}
vector<pair<int, int>> divide(vector<pair<int, int>> a)
{
if (a.size() <= 5)
return bruteHull(a);
vector<pair<int, int>>left, right;
for (int i=0; i<a.size()/2; i++)
left.push_back(a[i]);
for (int i=a.size()/2; i<a.size(); i++)
right.push_back(a[i]);
vector<pair<int, int>>left_hull = divide(left);
vector<pair<int, int>>right_hull = divide(right);

return merger(left_hull, right_hull);
}
int main()
vector<pair<int, int> > a;
a.push_back(make_pair(0, 0));

int n = a.size();
sort(a.begin(), a.end());
vector<pair<int, int> >ans = divide(a);
cout << "convex hull:\n";
for (auto e:ans)
cout << e.first << " "
<< e.second << endl;
return 0;
}
```