Convex Hull

```
process convex_hull(points[1...n])
Input: Array of points. Each point has a point.x, point.y
Output: lines
    points = sort_points(points)  //O(n log n)

    if n<=3: return points //base case for DandC (3,2 points)

    l,r = convex_hull(points[1...n//2]),
convex_hull(points[n//2+1...n])
    hull = merge(l,r)
    return hull

process merge(l,r):
    find_upper_tangent(l,r)

process find_upper_tangent(l,r):
    Input: two arrays of points each representing two polygons
    Output: a tuple with two points of a line
    l_idx = -1
    r_idx = 0
    slope = slope(l[l_idx],r[r_idx])
    is_left_tangent, is_right_tangent = False,False
    while(!is_left_tangent and !is_right_tangent):
        while(!is_left_tangent):
            new_slope = slope(l[l_idx-1], r[r_idx])
            if new_slope < slope: //if slope decreases, we've
found it
                is_left_tangent = True
            else:
                l_idx -= 1
            slope = new_slope
        while(!is_right_tangent):
            new_slope = slope(l[l_idx], r[r_idx+1])
            if new_slope < slope:
                is_right_tangent = True
            else:
                r_idx += 1
            slope = new_slope
    return (l[l_idx], r[r_idx])
```

```
process slope(p1,p2):
    return (p2.y-p1.y)/(p2.x,p1.x)

process sort_points(points[1...n]):
Input: Array of points. Each point has a point.x, point.y
if points[n].x > points[1].x:
    return
sort_helper(sort_points(points[1...n//2]),sort_points(points[n/
/2+1...n]))
else:
    return points


process sort_helper(u[1...k], v[1...l])
if k.x=0: return v
if l.x=0: return u
if u[1].x <= v[1].x
    return x[1].append(sort_helper(x[2...k],y[1...l]))
else:
    return y[1].append(sort_helper(x[1...k],y[2...]))
```