

算法设计与分析 C5-E

21377206 阮阳栋

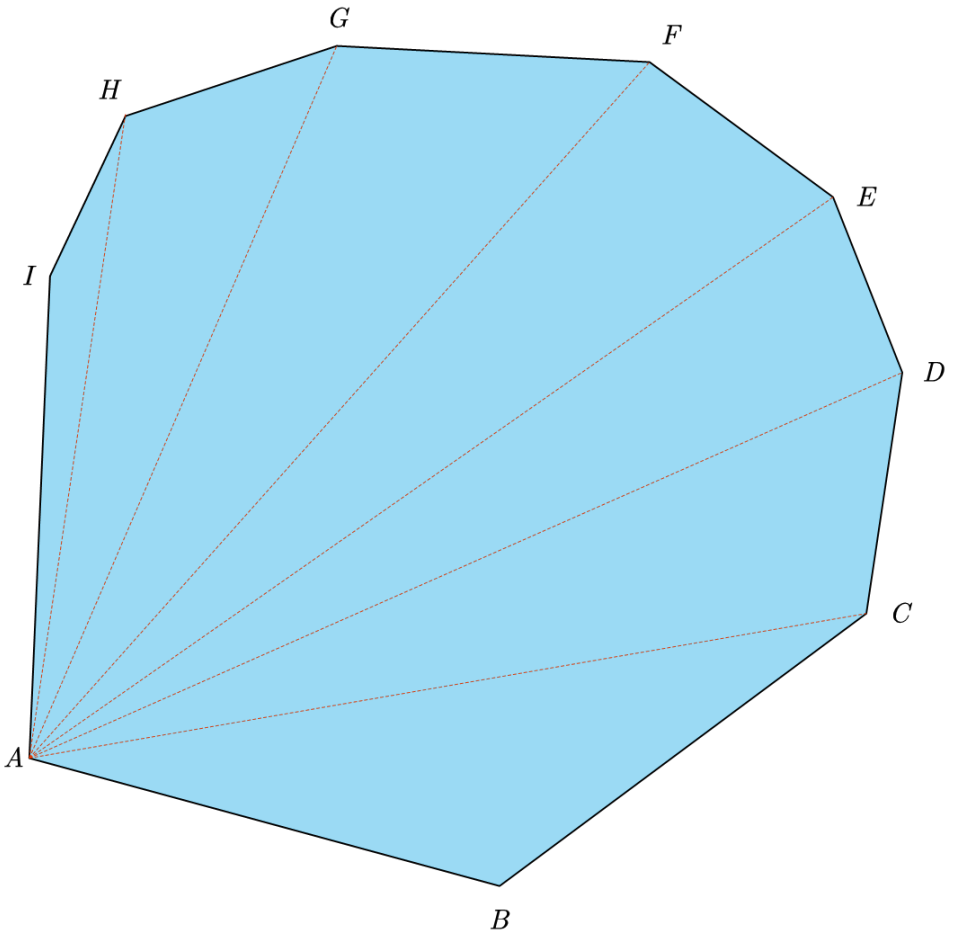
题目描述

给定 \mathbb{R}^2 平面上的点集 $S = \{P_1, P_2, \dots, P_n\}$, 求解凸包的面积。

题目分析

在点集数据分布不确定时，凸包算法采用Graham算法。

可以以左下点分割凸包求解凸包面积：



题目求解

使用Graham算法求解凸包：

```
void graham(){
    pt p0 = a[1]; int id = 1;
    for (int i=2;i<=n;i++){
        if (a[i].y < p0.y || (a[i].y == p0.y && a[i].x < p0.x))
            p0 = a[i], id = i;
    }
    a[id] = a[1], a[1] = p0;
    cpp_stablesort(2, n);

    for (int i=1;i<=n;i++){
        if (i > 2 && !direc(a[1], a[i-1], a[i])) continue;
        na[cnt++] = a[i];
    }
    cnt--;

    if (cnt-1 < 2) return;
    while (!s0.empty()) s0.pop();
    s0.push(na[1]), s0.push(na[2]), s0.push(na[3]);
    for (int i=4;i<=cnt;i++){
        while (1){
            pt iniTop = s0.top(), nextTop;
            s0.pop(), nextTop = s0.top(), s0.push(iniTop);
            if (direc(s0.top(), na[i], nextTop) >= 0) s0.pop();
            else break;
        }
        s0.push(na[i]);
    }
}
```

之后使用栈 s0 的元素计算面积：

```
ll result = 0;
//处理栈中元素到一个数组中（假定为p）
result += abs(direc(p[0], p[i], p[i+1]))
//使用叉积计算面积（ $ab \sin \theta$ ），最后一起除2
```

在输出时要注意，因为 double 精度不够，所以自己输出小数点：

```
if (result%2) printf("%lld.5", result/2);  
else printf("%lld.0", result/2);  
puts("");
```

时间复杂度

Graham算法运行时间为 $O(n \log n)$ ，其他部分算法均为线性时间复杂度，所以最终复杂度为 $O(n \log n)$ 。