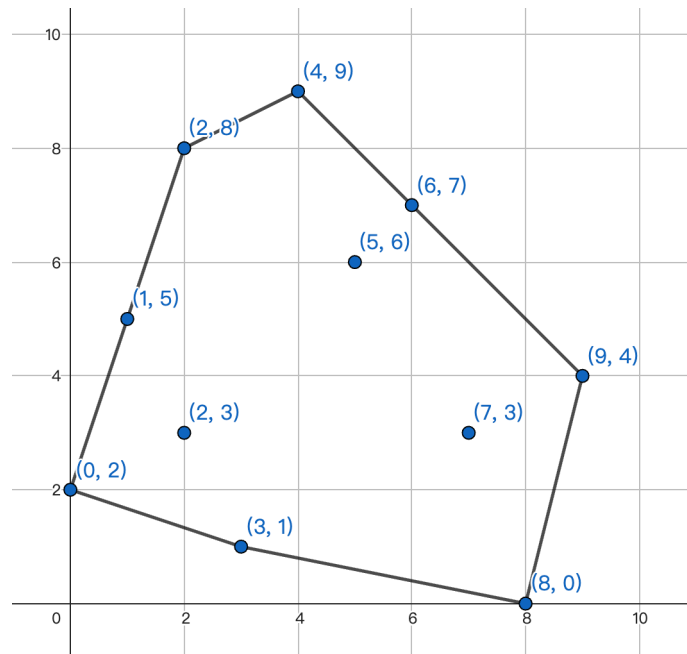


本次作业提交截止时间：2023年10月16日23:59

！ 注意事项

- 每次作业请上传至对应的文件夹，比如第一次作业上传至文件夹homework1。
 - 该文件夹中包含两个文件夹：code和pdf。
 - 非代码请上传pdf文件至文件夹pdf中。非pdf（比如doc文件、md文件、tex文件）请转换成pdf。
 - C++代码请上传至文件夹code中。本课程只允许C++代码。
 - pdf文件和cpp文件都用自己的学号命名。比如1001.pdf和1001.cpp。
 - 涉及公式的作业，推荐使用markdown编辑器。
 - 涉及算法伪代码的作业，推荐使用在线 LAT_EX 编辑器[Overleaf](#)完成，提交编译生成的pdf文档即可。如果需要`在overleaf中输入中文并能正确编译`，首先在文档中加入`package: \usepackage[UTF8]{ctex}`，然后修改设置：设置->修改Latex引擎->选择“XeLatex”。
 - 请基于提供的C++代码框架编写代码，注意：
 - 代码中不要包含任何中文
 - 不要使用`<bits/stdc++.h>`头文件
 - 不要在main函数后放置任何代码
 - 不要改变给定的函数原型
-

1. 如果在给定的有 n 个元素的多重集合中，某元素的出现次数大于 $n/4$ ，则称它是普遍的。设计一个 $O(n)$ 的算法来查找一个普遍的元素（如果存在）。给出算法的基本思路和伪代码描述，分析算法的时间复杂度。
2. 实现计算凸包的三种分治算法，即insert_hull, merge_hull, quick_hull。算法返回凸包边界上的点（顺时针方向，第一个点是具有最小 x 坐标的点）。下图给出了一个测试用例。对于该测试用例，你的算法应该返回(0,2), (2,8), (4,9), (9,4), (8,0), (3,1)。



```

1  // Note:
2  // You are free to utilize any C++ standard library functions.
3  // Please ensure to include the necessary headers below.
4  // Avoid using <bits/stdc++.h> to prevent potential compilation errors that could
   result in a score of zero.
5  #include <iostream>
6  #include <complex>
7  #include <vector>
8  #include <utility>
9  #include <list>
10
11  using namespace std;
12
13  typedef long double LD;
14  typedef complex<LD> Point;
15
16  // insert code here...
17
18
19  list<Point> insert_hull(vector<Point>& points)
20  {
21      // insert code here...
22  }
23
24
25  list<Point> merge_hull(vector<Point>& points)
26  {
27      // insert code here...
28  }
29
30  list<Point> quick_hull(vector<Point>& points)
31  {

```

```
32     // insert code here...
33 }
34
35
36 int main(int argc, const char * argv[]) {
37     // insert code here...
38     return 0;
39 }
40
41 // Please refrain from including any code beyond the main function,
42 // as any additional code will be removed during the code evaluation process.
```