HW3

Problem 1

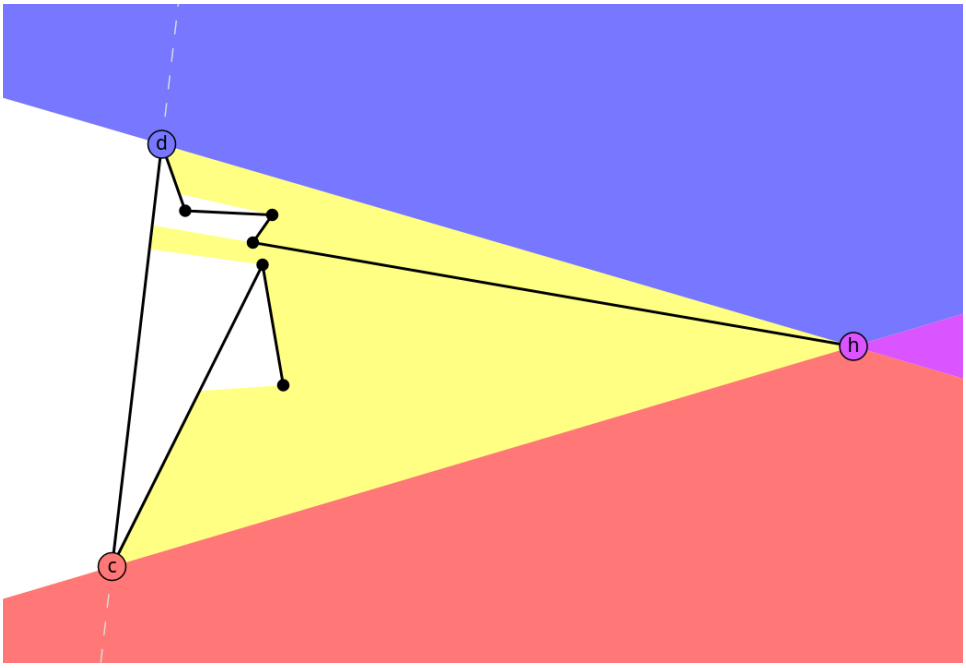
- 1.

Deque	Left test	Left test
<2,1,0,2>		
<3,2,1,0,3>	Left(2,1,3)T	Left(0,2,3)F
<3,2,1,0,3>	Left(3,2,4)T	Left(0,3,4)T
<5,3,2,1,0,5>	Left(3,2,5)T	Left(0,3,5)F
<6,3,2,1,0,5,6>	Left(5,3,6)F	Left(0,5,6)T
<6,3,2,1,0,5,6>	Left(6,3,7)T	Left(5,6,7)T
<6,3,2,1,0,5,6>	Left(6,3,8)T	Left(5,6,8)T
<9,2,1,0,5,6,9>	Left(6,3,9)F	Left(5,6,9)T
<10,1,0,5,6,9,10>	Left(9,2,10)F	Left(6,9,10)T

  the first and the last one of each deque are the bottom and the top.

  2.

- ◦ 1. Impossible, Edge v_4v_5 would be forced to cross the path $v_0v_1v_2$.
  ◦ 2. Possible,



  ◦ 3. Impossible, <2, 1, 0, 2> means 0 is on the left of v_2v_1. However, <4, 1, 2, 4> contains only right of v_2v_1, which does not include 0.

Problem 2

```c
#include <stdio.h>
#define true 1
#define false 0

typedef int tPointi[2];

int Area2(tPointi a, tPointi b, tPointi c) {
    return (b[0] - a[0]) * (c[1] - a[1]) -
            (c[0] - a[0]) * (b[1] - a[1]);
}

int Collinear(tPointi a, tPointi b, tPointi c) {
    return Area2(a, b, c) == 0;
}

int Between(tPointi a, tPointi b, tPointi c) {
    if (!Collinear(a, b, c))
        return false;

    if (a[0] != b[0])
        return ((a[0] < c[0] && b[0] > c[0]) ||
            (a[0] > c[0] && b[0] < c[0]));
    else
        return ((a[1] < c[1] && b[1] > c[1]) ||
            (a[1] > c[1] && b[1] < c[1]));
}

int TIntersect(tPointi a, tPointi b, tPointi c, tPointi d) {
    if (Between(a, b, c) && !Collinear(a, b, d))
        return true;

    if (Between(a, b, d) && !Collinear(a, b, c))
        return true;

    if (Between(c, d, a) && !Collinear(c, d, b))
        return true;

    if (Between(c, d, b) && !Collinear(c, d, a))
        return true;

    return false;
}

int main() {
    tPointi a = {0, 0};
    tPointi b = {0, 2};
    tPointi c = {0, 1};
```

```
    tPointi d = {1, 1};
    printf("TIntersect is %s", TIntersect(a, b, c, d)?"true":"false");
}
```