

## LAB-7

-Saloni Patel

### ➤ Convex Hull using Graham scan:

```
print("how many points you want to enter (enter in sorted order)")
n=(int)(input())
def direction(dim1,dim2,dim3):
    ans=((dim3[0]-dim1[0])*(dim2[1]-dim1[1]))-((dim2[0]-
dim1[0])*(dim3[1]-dim1[1]))
    return(ans)
l=[]
for _ in range(0,n):
    t=[int(x) for x in input().split(' ')]
    l.append(t)
l.append([0,0])
stack=[]
for i in range(0,3):
    stack.append(l[i])
for i in range(3,n+1):
    top=stack.pop()
    ntop=stack.pop()
    stack.append(ntop)
    stack.append(top)
    while(direction(ntop,top,l[i])>0):
        popo=stack.pop()
        top=stack.pop()
        ntop=stack.pop()
        stack.append(ntop)
        stack.append(top)
    stack.append(l[i])
stack.pop()
print("convex hull with points:",stack)
```

**snapshot:**

Jupyter lab7 Last Checkpoint: 4 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python

```
In [1]: print("how many points you want to enter (enter in sorted order)")
n=(int)(input())
def direction(dim1,dim2,dim3):
    ans=((dim3[0]-dim1[0])*(dim2[1]-dim1[1]))-((dim2[0]-dim1[0])*(dim3[1]-dim1[1]))
    return(ans)
l=[]
for _ in range(0,n):
    t=[int(x) for x in input().split(' ')]
    l.append(t)
l.append([0,0])
stack=[]
for i in range(0,3):
    stack.append(l[i])
for i in range(3,n+1):
    top=stack.pop()
    ntop=stack.pop()
    stack.append(ntop)
    stack.append(top)
    while(direction(ntop,top,l[i])>0):
        popo=stack.pop()
        top=stack.pop()
        ntop=stack.pop()
        stack.append(ntop)
        stack.append(top)
    stack.append(l[i])
stack.pop()
print("covex hull with points:",stack)

how many points you want to enter (enter in sorted order)
6
0 0
5 0
4 4
1 2
-1 3
covex hull with points: [[0, 0], [5, 0], [4, 4], [-1, 3]]
```

## Conclusion:

Running time:  $O(n \log n)$