

Assignment #4

201624548 이재윤

Intersection Rectangle Area

- 사용 언어
Python 3
- 사용 라이브러리
sys (Command Line input을 위해)
time (시간초를 재기 위해)
- 파일 분석

함수 내역

1. FindPoints
2. find_intersecting_rectangle_area
3. check_intersecting_rectangle_area
4. rec_area
5. sort_rectangle
6. main

아래 페이지부터 함수에 대한 소개 및 설명이 이어진다.

1. FindPoints

```
# x1 < x2, y1 < y2
# 겹치는 부분을 반환한다. ( 대각선 두 점 )
def FindPoints(rec1, rec2):
    x1, y1, x2, y2 = rec1
    x3, y3, x4, y4 = rec2

    intersecting_rec_x1 = max(x1, x3)
    intersecting_rec_y1 = max(y1, y3)

    intersecting_rec_x2 = min(x2, x4)
    intersecting_rec_y2 = min(y2, y4)

    # 겹치지 않는다면 return False
    if (intersecting_rec_x1 >= intersecting_rec_x2 or intersecting_rec_y1 >= intersecting_rec_y2) :
        return False

    # 겹치는 부분이 있다면 list로 변환하여 반환
    rec3 = [intersecting_rec_x1, intersecting_rec_y1, intersecting_rec_x2, intersecting_rec_y2]
    return rec3
```

max와 min을 사용하여 겹치는 사각형을 구한다.

겹치지 않는 부분이 있다면 return False

겹친다면 사각형을 리스트로 변환하여 반환한다.

2. find_intersecting_rectangle_area

```
# FindPoints 를 이용하여 겹치는 사각형을 반환한다
# 브루트포스
def find_intersecting_rectangle_area(Rectangle_list) :

    intersecting_rectangle = list()

    # 사각형이 겹치는지 확인하여 겹친다면 리스트에 삽입한다.
    for i in range(len(Rectangle_list)) :
        for j in range(i+1, len(Rectangle_list)) :
            result = FindPoints(Rectangle_list[i], Rectangle_list[j])

            if result != False and result not in intersecting_rectangle :
                intersecting_rectangle.append(result)

    return intersecting_rectangle
```

모든 사각형을 위 1번의 함수를 이용하여 겹치는 부분이 있는지 확인하여 list에 넣은 후 반환해준다. 이 list는 이후 3번의 함수에서 사용된다.

3. check_intersecting_rectangle_area

```
# find_intersecting_rectangle_area을 이용하여
# 모든 사각형의 겹치는 부분을 반환한다
def check_intersecting_rectangle_area(Rectangle_list) :

    intersecting_rectangle = find_intersecting_rectangle_area(Rectangle_list)
    result_area = rec_area(intersecting_rectangle)

    # 중복되어 겹치는 부분을 제거한다.
    # 중복되는 부분이 없을 때 까지 반복
    while intersecting_rectangle :
        intersecting_rectangle = find_intersecting_rectangle_area(intersecting_rectangle)
        result_area = result_area - rec_area(intersecting_rectangle)

    return result_area
```

가장 중요한 함수이다.

모든 사각형의 겹치는 영역의 넓이를 반환하는 함수이다.

중간 while문이 중복을 제거하는 함수인데,
중복 사각형들 내에서 중복되는 부분을 찾고
이를 계속해서 반복하며 area에서 빼 주게 된다면
중복이 모두 제거된 area가 나오게 된다.

이 area가 최종 목적 area가 되는 것이다.

이렇게 구한 area를 반환해준다.

4. rec_area

5, sort_rectangle

```
# 리스트 내 모든 사각형의 넓이를 반환
def rec_area(rec_list) :
    area = 0
    for x1,y1,x2,y2 in rec_list :
        area = area + abs(x1-x2) * abs(y1-y2)
    return area

# x1 < x2, y1 < y2
# list를 정렬하여 list로 반환
def sort_rectangle(tmplist) :
    x1, y1, x2, y2 = tmplist

    x1, x2 = min(x1,x2), max(x1,x2)
    y1, y2 = min(y1,y2), max(y1,y2)
    rec = [x1, y1, x2, y2]
    return rec
```

너무 단순한 내용이라 설명은 생략하겠다.

6. main

메인 함수이다. Input을 확인하고,
파일을 읽거나 쓰고, 출력한다

```
def main():

    if len(sys.argv) != 2 :
        print("Usage : PA04.py [INPUT FILE]")
        return

    start = time.time()

    Rectangle_list = list()
    filename = sys.argv[1].rstrip()
    with open(filename, 'r') as f :
        for l in f.readlines() :
            tmplist = list(map(int, l.split()))
            if len(tmplist) > 1 :
                Rectangle_list.append(sort_rectangle(tmplist))

    area = check_intersecting_rectangle_area(Rectangle_list)
    end = time.time() - start
    with open("output.txt", 'w+') as f :
        f.write("Intersection Area : " + str(area) + "\n")
        f.write("Elapsed time : " + str(end))

    print("Intersection Area : " + str(area))
    print("Elapsed time : " + str(end))

main()
```

텍스트의 길이를 확인하여 2 이상,
즉 사각형일 경우에만 list에 삽입한다.

다른 내용은 너무 단순한 내용이라 설명은 생략한다.

■ 결과

아래는 pdf 파일에서 제공해준 Sample을 가지고 테스트 해 본 결과이다.

약 0.00007초 가량이 나왔다.

```
lep@lep-Virtual-Machine: ~/HW/Algorithm/Assignment#4
File Edit View Search Terminal Help
lep@lep-Virtual-Machine:~/HW/Algorithm/Assignment#4$ ls
'Intersecting Rectangle.py'  Sample.txt  output.txt
lep@lep-Virtual-Machine:~/HW/Algorithm/Assignment#4$ python3 'Intersecting Rectangle.py' Sample.txt
Intersection Area : 20
Elapsed time : 7.05718994140625e-05
lep@lep-Virtual-Machine:~/HW/Algorithm/Assignment#4$ cat output.txt
Intersection Area : 20
Elapsed time : 7.05718994140625e-05
lep@lep-Virtual-Machine:~/HW/Algorithm/Assignment#4$
```