(1) 퀵 정렬 알고리즘

--------------------------------------------------------------------

quickSort(a[], l, r)

if (r > l) then {

i ← partition(a, l, r)

quickSort(a, l, i-1);

quickSort(a, i+1, r);

}

end quickSort()

partition(a[], l, r)

v ← a[r]; // 가장 오른쪽 원소를 피봇으로 정함

i ← l-1; // 왼쪽에서 오른쪽으로 움직이는 포인터

j ← r; // 오른쪽에서 왼쪽으로 움직이는 포인터

for ( ; ; ) do {

do { i ← i + 1; } while (a[i] < v);

do { j ← j - 1; } while (a[j] > v);

if (i ≥ j) then break;

a[i]와 a[j]를 교환;

}

a[i]와 a[r]를 교환;

return i;

end partition()

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def quickSort(a, l, r):

if r > l:

i = partition(a, l, r)

quickSort(a, l, i-1)

quickSort(a, i+1, r)

def partition(a, l, r):

v, i, j = a[r], l-1, r

while True:

i += 1

while a[i] < v:

i += 1

j -= 1

while a[j] > v:

j -= 1

if i >= j:

break

a[i], a[j] = a[j], a[i]

a[i], a[r] = a[r], a[i]

return i

N = 10

a = [-1, 6, 2, 8, 1, 3, 9, 4, 5, 10, 7]

quickSort(a, 1, N)

print(a)

--------------------------------------------------------------------

(2) 합병 정렬 알고리즘

--------------------------------------------------------------------

mergeSort(a[], l, r)

if (r > l) then {

m ← (r+l)/2;

mergeSort(a[], l, m);

mergeSort(a[], m+1, r);

merge(a[], l, m, r);

}

end mergeSort()

merge(a[], l, m, r)

i ← l; j ← m+1; k ← l;

while (i ≤ m and j ≤ r) do {

if (a[i] < a[j]) then {

b[k] ← a[i];

k ← k+1; i ← i+1;

}

else {

b[k] ← a[j];

k ← k+1; j ← j+1;

}

}

if (i > m) then

for (p ← j; p ≤ r; p ← p+1) do {

b[k] ← a[p];

k ← k+1;

}

else

for (p ← i; p ≤ m; p ← p+1) do {

b[k] ← a[p];

k ← k+1;

}

for (p ← l; p ≤ r; p ← p+1) do

a[p] ← b[p];

end merge()

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def mergeSort(a, l, r):

if r > l:

m = (r+l) // 2

mergeSort(a, l, m)

mergeSort(a, m+1, r)

merge(a, l, m, r)

def merge(a, l, m, r):

i, j, k = l, m+1, l

while i <= m and j <= r:

if a[i] < a[j]:

b[k] = a[i]

k += 1

i += 1

else:

b[k] = a[j]

k += 1

j += 1

if i > m:

for p in range(j, r+1):

b[k] = a[p]

k += 1

else:

for p in range(i, m+1):

b[k] = a[p]

k += 1

for p in range(l, r+1):

a[p] = b[p]

N = 10

a = [None, 6, 2, 8, 1, 3, 9, 4, 5, 10, 7]

b = a.copy()

mergeSort(a, 1, N)

print(a)

--------------------------------------------------------------------

(3) 히프 정렬 알고리즘

--------------------------------------------------------------------

heapSort(a[], n)

for (i ← n/2; i ≥ 1; i ← i-1) do

heapify(a, i, n);

for (i ← n-1; i ≥ 1; i ← i-1) do {

a[1]과 a[i+1]을 교환;

heapify(a, 1, i);

}

end heapSort()

heapify(a[], h, m)

v ← a[h];

for (j ← 2 × h; j ≤ m; j ← 2 × j) do {

if (j < m and a[j] < a[j+1]) then j ← j + 1;

if (v ≥ a[j]) then exit;

else a[j/2] ← a[j];

}

a[j/2] ← v;

end heapify()

--------------------------------------------------------------------

□ 파이썬 소스 코드

--------------------------------------------------------------------

def heapify(a, h, m):

v, j = a[h], 2 \* h

while j <= m:

if j < m and a[j] < a[j+1]:

j += 1

if v >= a[j]:

break

else:

a[j // 2] = a[j]

j \*= 2

a[j // 2] = v

def heapSort(a, n):

for i in range(int(n/2), 0, -1):

heapify(a, i, n)

for i in range(n-1, 0, -1):

a[1], a[i+1] = a[i+1], a[1]

heapify(a, 1, i)

N = 10

a = [None, 6, 2, 8, 1, 3, 9, 4, 5, 10, 7]

heapSort(a, N)

print(a)

--------------------------------------------------------------------

**<코딩 테스트 연습 #1> 구간 합 구하기**

□ 파이썬 소스 코드

--------------------------------------------------------------------

N, M = map(int, input().split())

info = [list(map(int, input().split())) for \_ in range(N)]

for i in range(N):

for j in range(1, N):

info[i][j] += info[i][j-1]

for i in range(1, N):

for j in range(N):

info[i][j] += info[i-1][j]

for i in range(M):

x1, y1, x2, y2 = map(int, input().split())

answer = info[x2-1][y2-1]

if x1 > 1:

answer -= info[x1-2][y2-1]

if y1 > 1:

answer -= info[x2-1][y1-2]

if x1 > 1 and y1 > 1:

answer += info[x1-2][y1-2]

print(answer)

--------------------------------------------------------------------