**프로그래밍 실습 #3**

2023년 9월 3주차

□ 다음 정렬 알고리즘을 ADL로 작성하고 파이썬으로 구현해 보라.

- 성능을 측정할 때 충분히 큰 N에 대해 N, 2N, 3N 등으로 입력 데이터의 개수를 증가시켜 가면서 실행 시간이 어떻게 변화하는지 측정함

- 또한, 랜덤 순서, 정렬된 순서, 역순으로 정렬된 순서로 데이터가 입력될 때의 실행 시간을 측정하여 입력 데이터의 순서에 얼마나 민감한지를 측정함

- 보고서를 만들 때 엑셀 등을 사용하여 실행 시간의 변화를 그래프로 나타냄

(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

|  |
| --- |
|  |

return i;

end partition()

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

(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;

// 배열 b는 함수 외부에서“b = a.copy()” 명령문을 사용하여

// 주어진다고 가정

// a[i]와 a[j]를 비교하여 작은 값을 b[k]에 저장

|  |
| --- |
|  |

for (p ← l; p ≤ r; p ← p+1) do

a[p] ← b[p];

end merge()

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

(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];

|  |
| --- |
|  |

end heapify()

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

**<코딩 테스트 연습 #1>**

**□ 구간 합 구하기**

[문제]

N×N 개의 수가 N×N 크기의 표에 채워져 있다. (x1, y1)부터 (x2, y2)까지 합을 구하는 프로그램을 작성하시오. (x, y)는 x행, y열을 의미한다.

예를 들어, N = 4이고, 표가 아래와 같이 채워져 있는 경우를 살펴보자.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 5 |
| 3 | 4 | 5 | 6 |
| 4 | 5 | 6 | 7 |

여기서 (2, 2)부터 (3, 4)까지 합을 구하면 3 + 4 + 5 + 4 + 5 + 6 = 27이고, (4, 4)부터 (4, 4)까지 합을 구하면 7이다.

표에 채워져 있는 수와 합을 구하는 연산이 주어졌을 때, 이를 처리하는 파이썬 프로그램을 작성하라.

※ 코딩 테스트 연습은 ADL을 작성하지 않아도 됨

[입력]

- 첫째 줄에 표의 크기 N과 합을 구해야 하는 횟수 M이 주어진다. (1 ≤ N ≤ 1024, 1 ≤ M ≤ 100,000)

- 둘째 줄부터 N개의 줄에는 표에 채워져 있는 수가 1행부터 차례대로 주어진다.

- 다음 M개의 줄에는 네 개의 정수 x1, y1, x2, y2 가 주어지며, (x1, y1)부터 (x2, y2)의 합을 구해 출력해야 한다.

- 표에 채워져 있는 수는 1,000보다 작거나 같은 자연수이다. (x1 ≤ x2, y1 ≤ y2)

* N, M과 행렬을 입력 받는 파이썬 코드는 다음과 같다.

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

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

print()

print(N, M)

a = []

for \_ in range(N):

b = list(map(int, input().split()))

a.append(b)

print(a)

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

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

4 3

1 2 3 4

2 3 4 5

3 4 5 6

4 5 6 7

4 3

[[1, 2, 3, 4], [2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7]]

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

[출력]

총 M줄에 걸쳐 (x1, y1)부터 (x2, y2)까지 합을 구해 출력한다.

[실행 예]

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

4 3

1 2 3 4

2 3 4 5

3 4 5 6

4 5 6 7

2 2 3 4

3 4 3 4

1 1 4 4

27

6

64

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

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

2 4

1 2

3 4

1 1 1 1

1 2 1 2

2 1 2 2

1 1 2 2

1

2

7

10

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