

Homework #1

(Introduction to Data Structures)

Due date: Mar 29, 2018

학번: 2014310407

이름: 이 준 혁(JunHyuk Lee)

1. 최대 경사 찾기

1.1. Solution

우선 for 문을 이용하여 각 경사(배열의 $i+1$ 번째 값에서 i 번째 값을 뺀 값)을 `slop_now`, 즉 현재 기울기에 저장했다.

그리고 만약 이 경사가 이전 경사와 같다면, 중복도를 나타내는 변수 `accmul` 을 1 증가시켰다. 아닌 경우, 전번 경사와 중복되지 않는다는 의미이므로 `accmul` 을 1 로 초기화 시켰다.

그리고 중복도 * 현재 기울기의 절댓값이 이전에 저장된 최댓값(`max` 변수)보다 큰 경우, 중복도 * 현재 기울기의 절댓값을 `max` 에 저장했다.

또한 현재 기울기는 다음 번에는 이전 기울기가 되므로, 이전 기울기 변수(`slop_prev`)에 현재 기울기를 저장했다.

모든 경사를 다 비교하면 `max` 에는 최대 경사 값이 남게 된다. 따라서 `max` 를 반환 시켰다.

1.2. Result (snapshot)

```
C:\WINDOWS\system32\cmd.exe
13
0 11 22 32 20 11 5 9 14 3 12 21 0
22
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
8
0 1 19 37 44 2 15 0
42
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
10
0 4 12 9 21 30 22 34 17 0
34
계속하려면 아무 키나 누르십시오 . . .
```

2. 가능한 경로의 수 찾기

2.1. Solution

각 위치까지 가능한 경로의 수를 값으로 가지는, 똑 같은 크기의 정수형 배열(mark)을 만들었다.
mark 는 다음과 같은 방식으로 값을 채웠다.

1. $\text{mark}(0, 0)$ 위치의 값은 1 이다.
2. $\text{mark}(i, j)$ 위치의 값을 0 으로 초기화한다.
3. 만약 $\text{mark}(i, j)$ 위치에 해당되는 경로의 값($s[i][j]$)이 0 이라면, 가는 길이 없다는 의미이므로 그대로 0 으로 둔다.
4. 만약 $\text{mark}(i, j)$ 위치에 해당되는 경로의 값($s[i][j]$)이 1 이라면, $(i-1, j)$ 위치까지 경로의 수(단, $i > 0$ 인 경우) 와 $(i, j-1)$ 위치까지 경로의 수(단, $j > 0$ 인 경우)를 더한다. 이 값들을 더한 것은 (i, j) 위치까지 경로의 수이다. (왜냐하면 오른쪽 혹은 아래쪽으로 밖에 움직일 수 없으므로) 즉, 식으로 쓰면 $\text{mark}(i, j) = \text{mark}(i-1, j) + \text{mark}(i, j-1)$ 이다.
5. 이 과정을 $0 \leq i \leq R-1, 0 \leq j \leq C-1$ 인 동안 반복한다. 이렇게 된 경우 $\text{mark}(R-1, C-1)$ 의 값은 가능한 총 경로의 경우의 수와 일치한다. 이 값을 반환한다.

2.2. Result (snapshot)

```
C:\WINDOWS\system32\cmd.exe
5 5
1 1 0 1 1
1 1 0 0 1
1 1 1 1 0
1 1 0 1 1
1 1 1 1 1
1 1
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
3 4
1 1 1 1
0 0 0 1
0 1 1 1
1
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
4 5
1 1 1 1 1
1 0 1 0 1
0 1 1 1 0
1 0 1 1 1
2
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
5 5
1 1 1 1 0
0 0 0 1 0
0 1 1 1 0
0 0 0 0 1
1 1 1 1 1
0
계속하려면 아무 키나 누르십시오 . . .
```

3. 두 직사각형의 겹치는 영역의 넓이 구하기

3.1. Solution

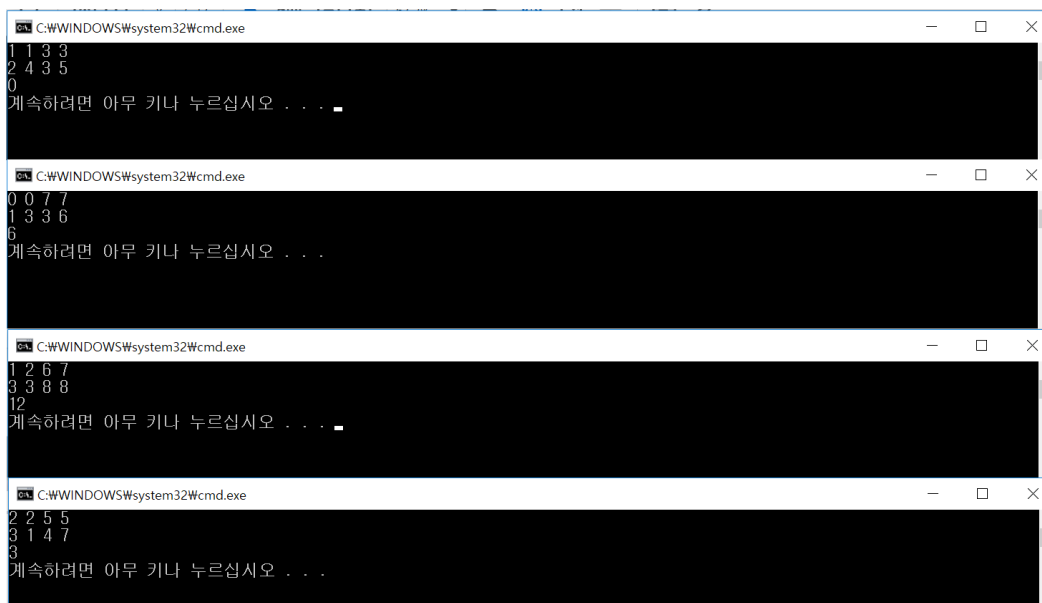
겹치는 직사각형의 넓이를 구하기 위해, 겹치는 직사각형의 밑변의 길이를 일반적으로 구하는 공식을 생각해 보았다.

두 직사각형이 겹칠 때 길이는 각 직사각형의 가로 길이를 두 개 더한 것에서 총 x 방향의 길이를 빼는 것이다. 즉, 직사각형 A, B에 대해 (A의 가로) + (B의 가로) - (x 방향 최댓값 - x 방향 최솟값)

만약 계산된 값이 음수가 나왔다면, (x 방향 최댓값 - x 방향 최솟값)이 (A의 가로) + (B의 가로)보다 크다는 의미이므로, 이는 직사각형들이 겹치지 않는다는 의미이다. 겹치는 직사각형이 없다는 의미, 즉 넓이가 0이라는 의미이므로 0을 반환했다. (세로 방향도 마찬가지로 음수일 때 0을 반환하기로 했다.)

만약 위에서 계산한 값이 가로 방향 뿐만 아니라 세로 방향으로도 양수가 나온다면, 이 직사각형은 분명히 겹치는 것이므로, 구한 두 값을 곱해서 겹친 직사각형의 넓이를 구한다.

3.2. Result (snapshot)



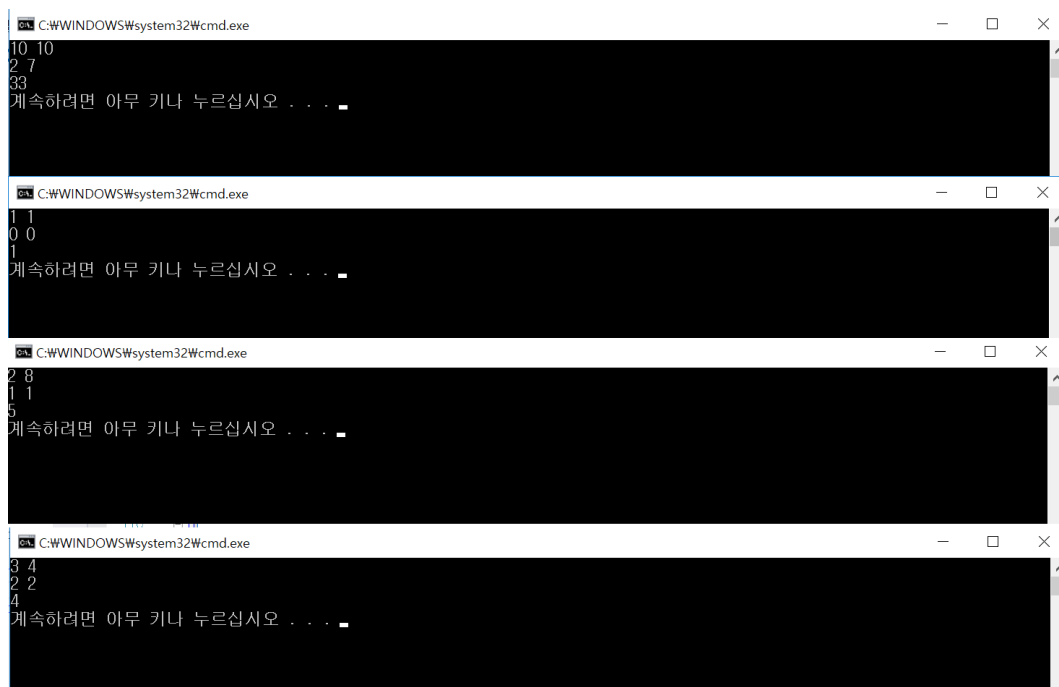
4. 가능한 돌의 수 세기

4.1. Solution

행의 수와 열의 수를 각각 R, C 라고 했을 때, $0 \leq a \leq R-1, 0 \leq b \leq C-1$ 인 a, b 에 대해, $(x-a)$ 의 절댓값과 $(y-b)$ 의 절댓값의 합을 3 으로 나눈 나머지가 3 인 a, b 의 개수를 세는 것 이다.

이를 위해 for 문을 두 번 사용해서, 모든 a, b 에 대해 $(x-a)$ 의 절댓값과 $(y-b)$ 의 절댓값의 합이 3 으로 나누어 떨어지면 cnt 를 1 씩 증가시켜서, cnt 를 반환했다. 이 cnt 의 값이 바로 가능한 돌의 수 이기 때문이다.

4.2. Result (snapshot)



5. 프리즈비 정렬하기

5.1. Solution

다음과 같은 아이디어로 정렬을 실시했다.

1. N 개의 수가 있으면, i 번째 수를 임시로 다른 변수에 저장한다. 단, $i \geq 2$

2. 앞의 수들 중에서, 임시로 저장한 수보다 더 작은 수가 나오기 전까지는 앞의 수들을 한 칸 씩 뒤로 당긴다.
3. 더 작은 수를 발견하면, 혹은 맨 처음까지 더 작은 수가 없으면, 그 다음 위치에 임시로 저장한 수를 넣는다.

우선 임시 프리즈비 변수(temp)를 만들었다. 그리고 i 번째 프리즈비를 temp 에 저장했다. 그리고 앞의 프리즈비의 지름(diameter)이 temp 프리즈비의 지름보다 크거나, 지름은 같은데 앞의 프리즈비의 이름이 알파벳 순으로 더 뒤에 위치하는 경우동안, 앞의 프리즈비들을 한 칸 씩 뒤로 이동시켰다.

이 때, 지름은 같은데 앞의 프리즈비의 이름이 알파벳상으로 더 앞인지 뒤인지 판별하기 위해 compare_string 함수를 구현했다.

이 함수는 두 문자열을 입력 받아, 각 문자열에서 '\0', 즉 NULL 문자를 발견하기 전까지 각 문자열의 i 번째 문자를 비교한다. 만약 이 때 단순히 아스키코드 값을 비교하면 안 되므로, alp_order 라는 함수를 사용했다.

alp_order 함수는 문자를 입력으로 받아 A 인 경우 1, a 인 경우 2, B 인 경우 3, ... 이런 식으로 $A < a < B < b < \dots$ 순으로 값을 반환해 주는 함수이다. 즉, 사전 순서에서 앞에 나오는 값이 더 작은 값으로 반환된다. 만약 입력이 알파벳이 아닌 경우 -1 을 반환한다.

이렇게 각 i 번째 문자를 비교해서, 첫 번째 문자열의 문자가 더 작은 경우에는(즉, 사전상으로 더 앞에 올 때) 1 을, 두 번째 문자열이 더 작은 경우는 -1 을 반환한다. 만약 i 번째 문자가 같으면 i 를 1 증가 시켜서 또 비교한다.

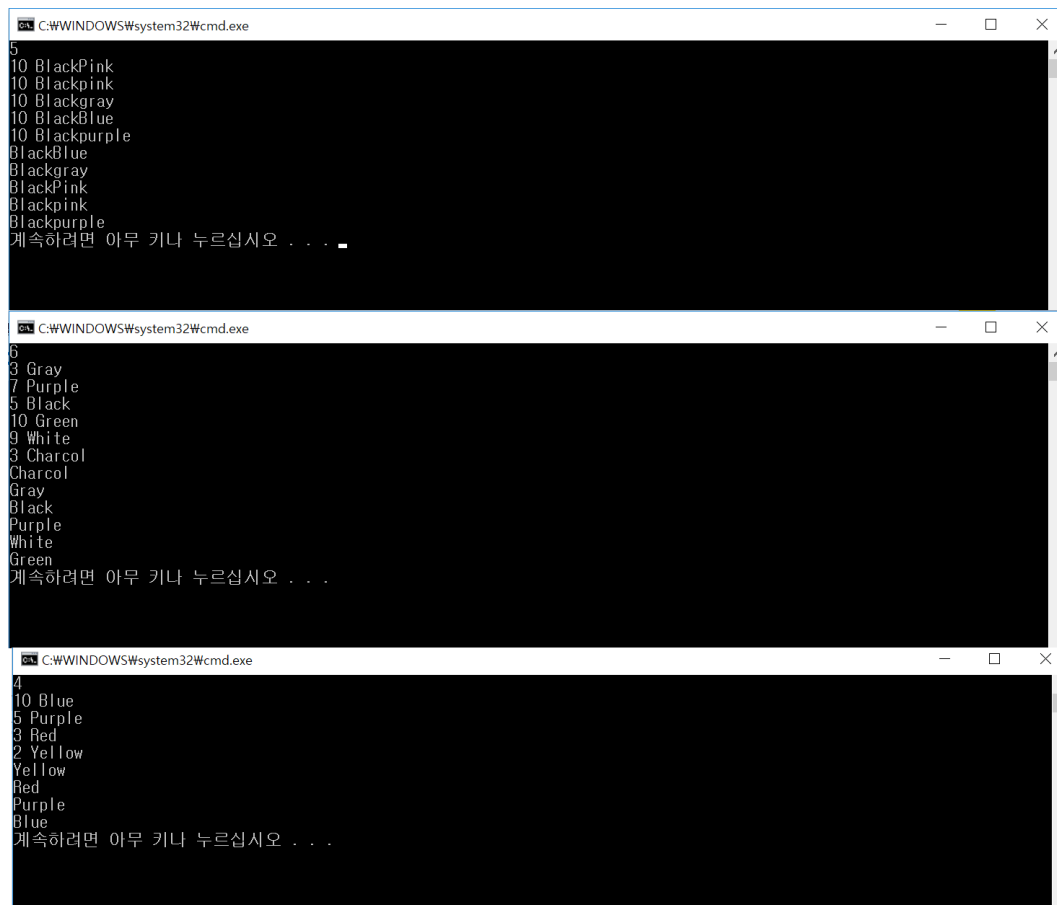
만약 NULL 문자가 나올 때 까지 반환 값이 없다면, 한 문자열이 다른 문자열에 포함되거나 같다는 의미이다. 이렇게 되면 두 문자열이 완전히 같은 지, 즉 둘 다 i 번째 문자가 NULL 문자인지를 확인한다. 이 경우에는 0 을 반환한다.

만약 첫 번째 문자열의 i 번째 문자만 NULL 문자라면 첫 문자가 더 짧고, 이 말은 사전상으로 더 먼저 온다는 말이므로 1 을 반환했다. 반대의 경우에는 -1 을 반환했다.

따라서 지름이 같은 경우에, 앞의 프리즈비가 임시로 저장된 프리즈비보다 사전상으로 앞에 있다면, 바로 종료하면 되므로 break 를 사용했다.

이 과정이 끝나면 모든 프리즈비가 정렬된 것이므로, 이를 화면에 출력했다.

5.2. Result (snapshot)



```
C:\WINDOWS\system32\cmd.exe
5
10 BlackPink
10 Blackpink
10 Blackgray
10 BlackBlue
10 Blackpurple
BlackBlue
Blackgray
BlackPink
Blackpink
Blackpurple
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
6
3 Gray
7 Purple
5 Black
10 Green
9 White
3 Charcol
Charcol
Gray
Black
Purple
White
Green
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
4
10 Blue
5 Purple
3 Red
2 Yellow
Yellow
Red
Purple
Blue
계속하려면 아무 키나 누르십시오 . . .
```