2025학년도 1학기 알고리즘 (학수번호: CSC2008-03)

알고리즘 설계문제

담 당 교 수 : 주 종 화

학 번 : 2022110151

이 름 : 이 주 연

1. [설계문제] 다음 요구사항을 해결하기 위한 알고리즘을 설계하고자 한다.

입력: 2차원 좌표평면 상의 직사각형 1개와 원 1개

출력: 직사각형이 원에 완전히 포함되거나 접하는 경우 1을 출력

원이 직사각형에 완전히 포함되거나 접하는 경우 –1을 출력

그렇지 않은 경우 0을 출력

아래 각 단계에 맞추어 알고리즘을 설계하시오. (총 20점, 부분점수허용)

[참고] 점 (x1, y1)과 직선 ax+by+c=0 사이의 거리 d = $|ax1 + by1 + c| / (a^2 + b^2)^{1/2}$

(1) **2차원 좌표평면 상에서 직사각형, 원 각각의 위치 및 형태를 입력받기 위한 적절한 자료구조를 제안하시오. 또한, 이를 통해 전체 알고리즘의 입력 형태를 구체화하시오. (5점)**

```
typedef struct {
        int x1, y1;   // 4개의 점 중 왼쪽 아래에 있는 꼭짓점
        int x2, y2;   // 4개의 저 중 오른쪽 위에 있는 꼭짓점
} Rectangle;


typedef struct {
        int x, y;        // 원의 중심 x, y, 좌표
        int r;           // 반지름 r
} Circle;
```

2차원 좌표평면 상에서 직사각형, 원 각각의 위치 및 형태를 표현할 때, 직사각형은 두 꼭짓점(x1, y1), (x2, y2)로, 원은 중심 좌표(x, y)와 반지름 r로 표현할 수 있습니다. 직사각형의 경우, x1, x2, y1, y2로 직사각형의 네 꼭짓점을 모두 표현할 수 있으므로 이를 반영해 위와 같이 구조체(rectangle과 circle)를 정의하였습니다.

(2) (1)의 자료구조에 기초해 주어진 요구사항을 해결하기 위한 알고리즘을 작성하시오. 단, 알고리즘의 표현방법은 자연어, pseudo code, C/C++ code 중 임의로 선택해도 무방하며, 문제에서 명확히 제시하지 않은 부분에 대해서는 해당 내용을 명시한 뒤 합리적인 범위 내 에서 설계자가 풀이하기 좋은 방향으로 임의로 가정해도 무방하다. (각 5점, 10점)

1. 직사각형 좌표(x1, y1, x2, y2)와 원의 중심좌표(x,y)&반지름 r을 입력받는다.
2. checkRelation함수 호출을 통해 isCircleIn()함수 조건과 isRecIn()함수 조건을 체크한다.
3. 원이 직사각형에 포함되거나 접하는지 판별하는(isCircleIn()) 단계에서
   A. 먼저 원의 모든 경계가 직사각형 안에 모두 들어오는지 체크해, 모두 들어올 경우 -1 을 반환하고(원이 직사각형 안에 완전히 포함됨을 의미),
   B. 외부에서 직사각형과 접하는지 판별하기 위해, 원의 중심으로부터 가장 가까운 직사각형 점을 계산해서 이 점과 원 중심 사이의 거리 제곱이 r^2과 같은지 체크해, 같을 경우 -1 반환(접함을 의미)
   C. 내부에서 직사각형과 접하는지 판별하기 위해, 원 중심이 직사각형 안에 있고, 원 경계가 직사각형 변과 완전히 닿아 중심좌표에서 반지름을 더하고 뺀 값이 직사각형 변과 거의 같으면 -1 반환(접함을 의미)
   D. 이외에는 0 반환 (포함되지도, 접하지도 않음을 의미)
4. 직사각형이 원에 포함되거나 접하는지 판별하는(isRecIn()) 단계에서는
   A. 직사각형의 네 꼭짓점 좌표를 vertices 배열로 저장하고((x1,y1), (x1,y2), (x2,y1), (x2,y2)),
   B. 각 꼭짓점으로 반복문을 시행해, 해당 점에서 원 중심까지의 거리의 제곱을 구해
   C. 만약 어느 한 점이라도 거리의 제곱이 반지름의 제곱보다 클 경우 0을 반환하고(직사각형이 원 밖으로 벗어났다고 간주하기)
   D. 모든 점이 원 내부 혹은 접할 경우에는 1을 반환한다.

가독성을 높이기 위해 전체 알고리즘 구조는 자연어로 표현하고, 코드로도 확인하실 수 있도록 주석을 달아 circleAndRecRelation.c 파일도 함께 첨부했습니다.

(3) (2)의 알고리즘에 대한 시간복잡도를 빅세타 표기법으로 논하시오. (5점)

이 알고리즘의 최종 시간 복잡도는 Θ(1)입니다.

먼저, Rectangle, Circle의 고정된 구조체를 입력 받고 있고, 각 함수 내에서 상수 번씩 비교 연산을 진행합니다. 예를 들어 isCircleIn(Rectange r, Circle c)에서 경계에 포함여부를

확인하는데 4회, 가장 가까운 점을 설정하는데 2회, 거리 계산하는데 1회 비교 연산을 진행하고, 거리의 제곱 함수를 호출할 때 덧셈과 곱셈 연산만 포함되기 때문에 시간복잡도가 상수시간 Θ(1)이 나옵니다. isRectIn함수에서도 직사각형의 네 꼭짓점 각각에 대해서 원의 중심까지의 거리의 제곱을 구하고 이를 비교하는 작업을 총 4번 반복 수행하므로 이 또한 상수시간에 해당되빈다. 이와 같이 모든 조건 검사가 고정된 횟수의 연산과 비교로 진행되고, 입력 데이터 크기와는 상관없이 항상 일정한 시간 내에 종료되므로 알고리즘의 전체 시간 복잡도는 Θ(1)입니다. Θ(1) 시간 복잡도를 통해 직사각형과 원의 포함 관계 및 접함 여부를 판별할 수 있습니다.