

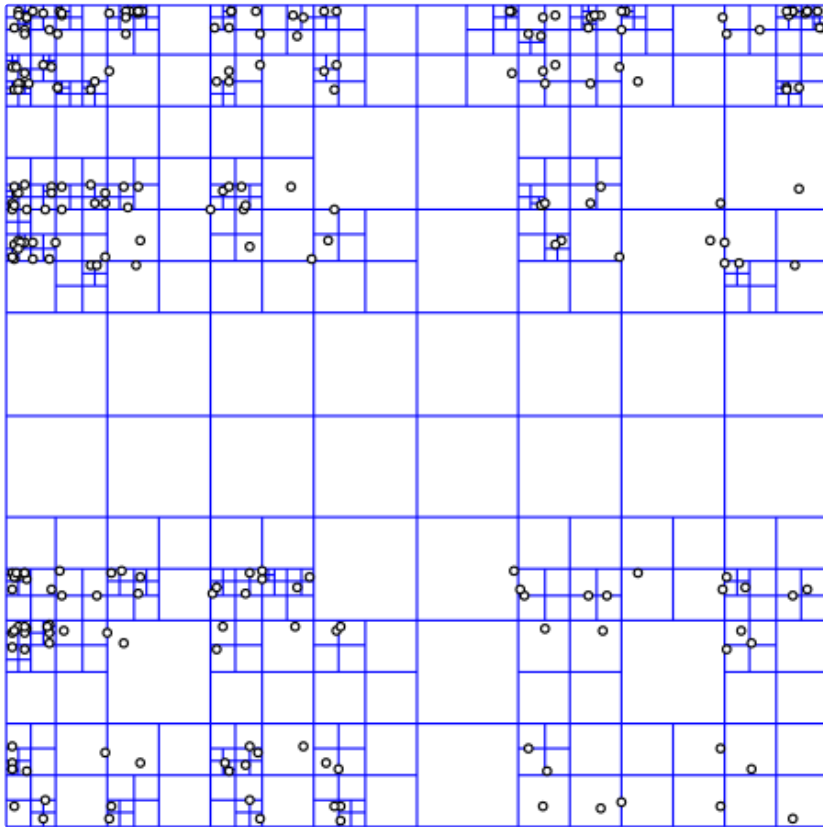
## ILE3-017 다차원 Tree

배열 형태의 1차원 데이터가 아닌, 공간 정보를 표현할 수 있는 2차원 이상의 정보를 저장할 수 있는 Tree

### Quad Tree [↗](#)

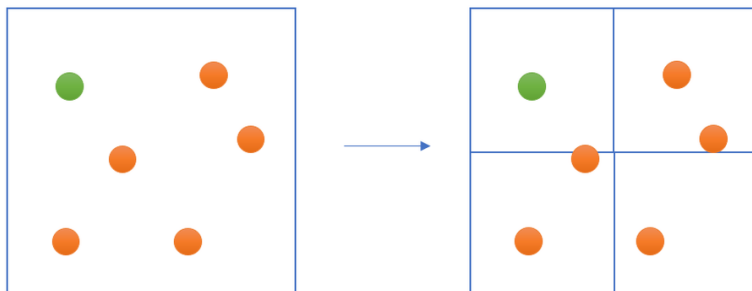
자식 노드를 4개를 지니는 트리

2차원 데이터에 대한 Index을 수행 할 때 가장 간단하게 사용 할 수 있는 구조

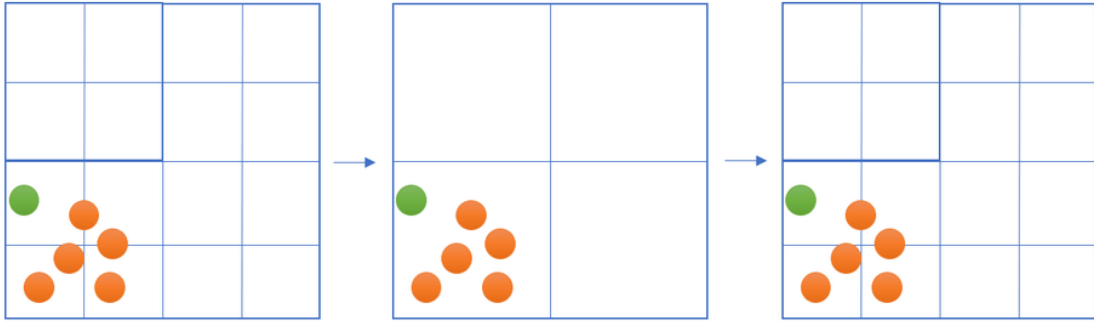


### INSERT [↗](#)

1. Root Node를 선택
2. 선택된 Node가 자식 노드를 지닌 경우
  - a. 데이터가 해당되는 자식 노드를 현재 노드로 선택하고 2.로 돌아감
3. 데이터(Point)를 현재 Node에 삽입
4. 삽입 후 Node 데이터의 개수가 일정 수치 이상 인 경우, 4개의 자식 노드를 생성하고, 영역에 따라 데이터를 이동



a. 새로 생성된 자식 Node의 데이터의 개수가 일정 수치가 이상 인 경우, 현재 Node를 해당 자식 Node로 변경하고 b.로 돌아감



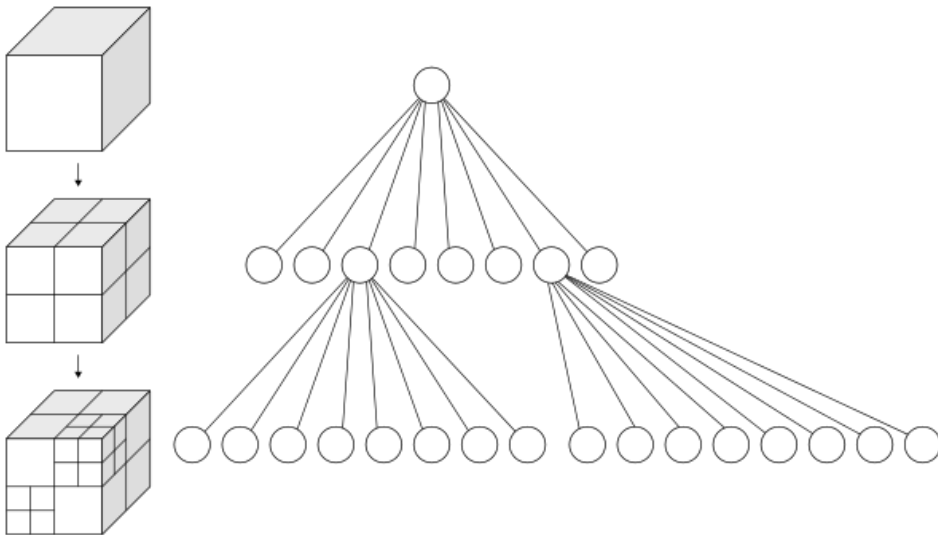
## GET [↗](#)

1. Root Node를 선택
2. 선택된 Node가 자식 노드를 지닌 경우
  - a. 선택할 범위에 해당하는 자식 노드를 선택하고 2.로 돌아감
3. Node 내의 데이터와 해당 값을 확인

## Octree [↗](#)

자식 노드를 8개 지니는 트리

3차원 데이터에 대한 Indexing을 수행 할 때 가장 간단하게 사용 할 수 있는 구조



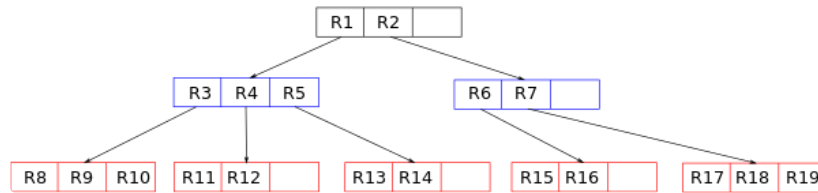
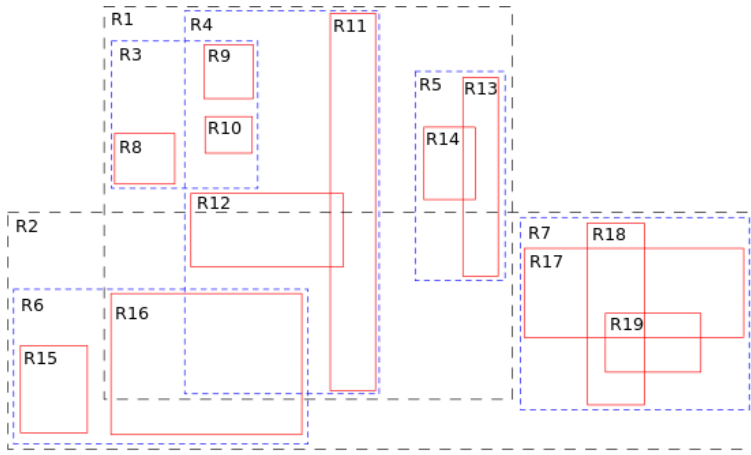
## R-Tree [↗](#)

최소 사각형을 이용하여 2차원 공간 데이터를 저장하는 방법

Leaf Tree에 대한 Level 및 개수가 일정하게 유지 할 수 있음

최소 사각형에 대한 중첩을 허용

Spatial Index에 가장 많이 사용되며, 변종이 다수 존재 (R\* Tree, R+ Tree)



## 과제 [🔗](#)

Quad Tree를 만들고, 주변에 존재하는 점의 개수를 측정하는 프로그램을 만들어 봅시다

## Quad Tree [🔗](#)

1. Root Node는 -180, -90, +180, +90의 범위를 표현합니다.
2. Quad Tree는 Insert, Get 기능을 지원하며, 2차원 Point 정보를 저장합니다 (lon, lat)
3. 각 Node는 가지고 있는 데이터가 16개를 초과 하는 경우 자식을 생성합니다.

## Test Program [🔗](#)

1. 랜덤으로 -180, -90, +180, +90 범위의 점 10,000개를 생성하여 List에 저장합니다.
2. List에 저장된 점에서 (127, 37)에서 거리가 1인 점의 개수를 출력하고 소요 시간을 측정합니다.
3. Quad Tree에 List에 저장된 점을 모두 추가하고, 동일하게 (127,37)에서 거리가 1인 점의 개수를 출력하고 소요 시간을 측정합니다.