**거리기반 여행 정보 서비스**

**적용 알고리즘** : merge Sort, Haversine

**알고리즘 개요**

**Merge Sort**

합병정렬 알고리즘은 분할정복 알고리즘의 일종으로 리스트를 절반으로 분할하고 사이즈가 1이 되면 분할 되어진 요소를 정렬 한 뒤 다시 재 조립하여 전체 리스트를 정렬하는 알고리즘이다.

합병정렬은 안정적인 정렬 알고리즘으로 성능과 안전성 모두 우수할 정렬 알고리즘 중 하나다. 합병정렬의 의사코드는 아래와 같다.

이번 서비스는 여행지 정보가 늘어난다면 리스트에 사이즈가 크다고 판단했다. 합병 정렬은 대용량 데이터를 정렬할 때 사용하기 좋기에 이번 거리기반 여행 정보 서비스 개발에 있어 적절하다고 판단했다.

**Haversine**

Haversine 알고리즘은 두 지점간의 직선 거리를 구하는데 사용되는 알고리즘 중 하나이다.

라이브 시간에 배운 멘헤튼 알고리즘은 직각 좌표계에서 사용되는 반면 실제 위도 경도는 지구 곡률을 고려해야한다는 점에서 우리 서비스에는 적절하지 않다고 판단하였다.

하서 사인 알고리즘은 구면 삼각법의 일종으로 지구의 곡률을 고려하며 두 지점 사이의 거리를 계산하는 알고리즘은 두 지점의 위도와 경도를 활용하여 거리를 계산하는데 수학적 식은 아래와 같다.

*a = sin²(Δlat/2) + cos(lat₁) \* cos(lat₂) \* sin²(Δlon/2)*

*c = 2 \* atan2(√a, √(1-a))*

*d = R \* c*

이러한 방식은 현재 GPS 기반 어플리케이션에서 주료 사용되는 공식이기에 이번 서비스에 적합하다고 판단했다.

**적용 서비스**

**내 위치 기반 관광지 정보 제공**

**적용 서비스 개발 개요**

사용자가 시도별 종류별 관광지역을 조회하면 데이터베이스를 통해 관광지 정보를 가져온다.

하지만 어떤 정보를 사용자에게 우선적으로 가져와야 고객이 만족할지에 대해 고민해본 결과 고객과 가까운 리스트를 보여주는 것이 높은 고객 만족도를 달성할 수 있다고 생각했다.

사용자의 위치는 Geolocation API를 활용하였고, 관광지 정보는 데이터베이스에 존재하는 위도와 경도를 활용하였다. 두 위도와 경도간의 거리를 구하는 로직은 Haversine 알고리즘을 사용하여 각 위도와 경도의 거리를 입력으로 받아 얼마나 떨어져 있는지를 *km*로 변환한다. 이러한 거리를 기존 객체에 wrapper하여 별도의 객체로 저장한다.

이 객체를 기준으로 sort 알고리즘인 merge sort를 사용한다.

Merge sort를 사용한 이유는 아래와 같다.

첫째, 효율적인 성능 어떠한 경우에서 O(n log n)의 시간복잡도를 가져 많은 관광지 정렬에도 우수한 성능을 보장한다.

둘째, 안정적 객체를 변형하지 않고 분할하는 방식이기에 기존 객체 Attraction의 불변성이 보장된다.

셋째, 동시성 가능 추후 실제 사용자들에게 배포 될 경우 요소를 분합하고 병렬하는 과정을 병렬로 처리하기 용이하다.

이를 통해 안정적으로 위치기반 정렬된 결과를 사용자는 제공 받을 수 있다.