**Enjoy Trip  
알고리즘 적용 보고서**

**서울 16반 김지원, 이준학**

**1. 퀵 정렬(Quick Sort) 알고리즘을 이용한 관광지 정보 조회**

1. **적용 알고리즘**

퀵 정렬(quick sort) 알고리즘

1. **알고리즘 개요**

퀵 정렬 알고리즘은 분할 정복(Divide and Conquer) 알고리즘 중 하나로, 다음과 같은 세 가지 주요 동작으로 나눌 수 있다.

분할 (Partitioning): 배열에서 하나의 원소를 선택하여 pivot(피봇)으로 설정한다. 피봇을 기준으로 작은 원소는 왼쪽으로, 큰 원소는 오른쪽으로 분할합니다. 이 과정을 피봇의 위치가 결정될 때까지 반복한다.

정복 (Recursion): 분할된 부분 배열에 대해 재귀적으로 퀵 정렬을 수행한다. 각 부분 배열은 독립적으로 정렬된다.

결합 (Combination): 부분 배열이 정렬되면, 이를 합쳐서 전체 배열을 정렬한다. 피봇의 위치가 중요하며, 피봇은 이미 정렬된 위치에 있게 된다.

퀵 정렬은 평균적으로 O(nlogn)의 시간 복잡도를 가지며, 분할된 부분 배열이 균등하게 분할되는 경우에 효율적으로 동작한다. 그러나 최악의 경우(피봇이 항상 최대 또는 최소값일 때)에는 시간 복잡도가 O(n^2)이 될 수 있으므로 피봇 선택 전략을 최적화해야 한다. 퀵 정렬은 일반적으로 빠르고 효율적인 정렬 알고리즘 중 하나로 널리 사용된다. 이러한 퀵 정렬 알고리즘을 활용하여 관광지 정보를 효율적으로 정렬하고, 사용자가 원하는 방식으로 조회하는데 유용한 솔루션을 개발할 수 있다.

1. **적용 서비스**

전국 관광지 정보 검색 결과 정렬

1. **적용 서비스 개요**

[지역 별 관광지] 페이지에서 전국 관광지 정보를 검색하고, 검색 결과를 관광지명, 주소, 위도, 경도 등의 기준에 따라 오름차순으로 정렬하여 사용자에게 제공한다.

검색 결과 화면에서 해당 테이블 헤더를 클릭하면 해당 열에 대한 오름차순 정렬된 결과를 표시합니다.

**관광지명 / 주소 / 위도 / 경도로 정렬한 화면**

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**알고리즘 구현 코드**

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**2. SHA-256 해시 알고리즘을 이용한 비밀번호 암호화 서비스**

1. **적용 알고리즘**

SHA-256 해시 알고리즘

1. **알고리즘 개요**
   1. SHA-256은 Secure Hash Algorithm(SHA)의 한 종류로, 입력값으로부터 고정 길이의 해시값을 생성하는 알고리즘이다. 256비트 길이의 해시값을 생성하며, 입력값의 길이와 관계없이 항상 고정된 길이의 해시값을 생성한다. 대칭키 암호화 방식으로 분류되며, 단방향 해시 함수이므로, 입력값에서 출력값으로의 변환은 가능하지만 출력값에서 입력값으로의 역변환이 불가능하다. 해시 함수는 입력값이 같으면 항상 같은 출력값을 반환하기 때문에 해시값과 원본 문자열 사이에 대응 관계가 있다. 이 대응관계를 미리 계산해 대량으로 저장해둔 레인보우 테이블을 사용해 공격한다면 해시 함수를 통한 보안 절차가 무력화 될 수 있다. 이러한 공격을 예방하고자 난수생성기로 만들어진 임의의 문자열인 Salt와 SHA-256 알고리즘을 함께 사용한다. 따라서 SHA-256 알고리즘을 이용하여 비밀번호를 암호화하면, 비밀번호를 평문으로 저장하는 것보다 안전하게 보호할 수 있다.
2. **적용 서비스**

회원가입, 로그인

1. **적용서비스 개발 개요**
   1. EncryptUtil 클래스를 작성하여 비밀번호 암호화를 구현하였다.
      1. a. createSalt() : 난수생성기를 사용하여 10자 길이의 Salt값을 반환하는 메소드
      2. getEncrypt() : 비밀번호와 Salt를 조합한 문자열을 SHA-256 알고리즘을 사용하여 암호화 하는 메소드
   2. EncryptUtil 인스턴스를 이용하여 회원가입, 로그인 서비스에서 각각 비밀번호 암호화를 진행하였다.
      1. 회원가입 : 회원에 대한 Salt값과 암호화된 알고리즘을 회원 table에 저장한다.
      2. 로그인 : 입력된 ID에 대한 회원의 정보를 불러와서 MemberDto 객체에 저장한다. 입력된 비밀번호와 회원의 Salt를 암호화하여 저장되어 있는 암호화된 비밀번호와 비교 후 동일하다면 회원 정보를, 아니라면 null 값을 반환한다.



