




03.19 - 관통프로젝트_알고리즘

!? HappyHouse 프로젝트에 알고리즘 적용

난이도	구현 기능	점수	비고
	알고리즘을 2개	50점	
	알고리즘 추가당	5점~20점	추가당 5
추가	프로젝트 요구사항 정의서	20점	
심화	프로젝트 화면 설계서	10점	

- 어떤 알고리즘을 사용하여 기능을 구현할지에 대한 계획 수립
- ex) 공지사항을 정렬하는 알고리즘
- 알고리즘 적용부분과 적용 내용을 명시한 기획서 (알고리즘의 장단점 포함)

관통프로젝트: HappyHouse_Algo_서울_08반_김정윤_김지원

 제출일: 2021.03.19

 참여 페어

- 김지원, 김정윤

처리된 요구사항 목록

난이도	구현기능	세부	작성여부 (O/X)	내용
기본	알고리즘 1	Merge Sort	O	검색 결과를 정렬하는 기능
기본	알고리즘 2	평균 필터 알고리즘	O	거래 시세 평균을 계산
기본	추가 알고리즘 3	Hash Algorithm	O	해당 아파트에 주변 상권(편의시설) & 환경정보 저장
기본	추가 알고리즘 4	KMP Algorithm	O	검색 bar에 입력한 keyWord로 전체 contents 검색
기본	추가 알고리즘 5	CBF Algoritm	O	찜 내역과 비슷하게 매물 추천 기능
기본	추가 알고리즘 6	Quick Sort	O	메인 페이지 차트 구현
추가	프로젝트 요구사항 정의서		O	
심화	프로젝트 화면 설계서		O	

▲ 주의 사항

- 작성된 기능은 반드시 캡처되어야 합니다.(GUI 실행화면, 콘솔 출력 등)
- 추가로 구현한 기능을 표에 추가시키세요.

1. 검색 정보 정렬기능 (Merge Sort 알고리즘)

적용 부분

1. 지역별 매물 결과 페이지
2. 아파트별 매물 결과 페이지

구현 목적

- 거래 매물 결과 정렬 기능을 추가하기 위함
 - 기본 : 최신순 (거래 날짜별로 정렬)
 - 지역별 / 아파트별 공통 정렬 기능 : 거리순, 시세순
 - 아파트별 상세 정보 정렬 기능 : 면적순, 층수 순, 환경 우수 순, 상권 우수 순
- 기본으로 최신순으로 정렬하고 거리순, 시세순으로 검색 시 동일한 결과를 가지는 것은 최신순으로 정렬되도록 구현

! ? 알고리즘 간단 설명

- 데이터가 문자열이기 때문에 Reference Type인 String 객체이므로 Merge Sort 알고리즘을 이용해서 날짜 별, 거리별, 시세별로 정렬한다.
- 퀵 정렬과 비슷하게 원본 배열을 절반씩 분할해가면서 정렬하는 정렬법
- 시간 복잡도는 $O(N \times \log(N))$

장단점

- 장점 : 퀵 정렬과 달리 기준값을 설정하는 과정 없이 무조건 절반으로 분할하기 때문에 기준값에 따라 성능이 달라지는 경우가 없음

따라서 항상 $O(N \times \log(N))$ 의 시간 복잡도를 가짐

- 단점 : 병합 정렬은 임시 배열에 원본 맵을 계속해서 옮겨주며 정렬을 하는 방식이므로 추가적인 메모리 필요
추가적인 메모리 할당을 할 수 없을 경우 병합정렬 사용 불가함

2. 거래 시세 평균 계산 (평균 필터 알고리즘)

적용 부분

1. 전국 매매가 동향
2. 지역별 평균 매매가 순위
3. 지역별 매물에서 아파트별 평균 매매가

구현 목적

- 지역별 아파트 정보의 시세는 해당 아파트의 전체 거래가의 평균을 보여주는 기능
- 지역별 각 아파트에 해당하는 모든 거래 가격을 더한 후 해당 갯수로 나누어 평균 거래 가격을 아래 매매가 위치에 표시
- 전국 거래동향 차트 구현시 전국 아파트 거래 동향을 각 날짜별 (월, 년도) 평균을 구한 후 차트에 출력하기 위함
- 지역별 거래동향 차트 구현시 각 지역의 아파트 거래 동향의 평균을 구해 차트에 출력하기 위함

1번 아파트

기준거리	300m
아파트이름	반포자이
전용면적	105m ²
매매가(단위 : 만)	290,000

!? 알고리즘 간단 설명

- n 번째 데이터가 들어온 데이터 셋의 평균을 구할 때 $n-1$ 까지의 평균을 재사용할 수 있는 알고리즘
- 이전에 구한 평균값을 이용하여 새로 들어온 데이터가 있을 경우 가중치 값을 이용하여 전체 평균 재계산
- 즉, for문을 계속해서 돌려서 전체 데이터의 합을 구한 후 그 합을 데이터의 갯수로 나누는 연산을 하지 않음

장단점

- 장점 : 기존의 평균을 구하는 계산을 할 경우 시간 복잡도는 $O(N)$ 이었는데 평균 필터 알고리즘을 사용하면 시간 복잡도는 데이터가 늘어나도 일정하게 $O(1)$ 의 시간 복잡도를 가짐 -> 매우 빠른 연산
- 단점 : 가중치를 계산하는 과정에서 이해하는데 어려움이 있고 계산식이 복잡함

3. 생활편의지수/환경오염지수 계산을 위한 자료구조 (HashMap)

적용 부분

1. 관심지역 환경오염지수 계산
2. 관심지역 생활편의지수 계산

구현 목적

- 관심지역의 생활편의지수와 환경오염지수를 수치화하고 해당 정보를 사용자들이 한눈에 알아볼 수 있게 제공하기 위함
- ex) 생활 점수 : 마트 1개당 +1, 학원 1개당 +1, 카페 1개당 +1
- ex) 대기 오염 점수 : 대기오염기업 1개당 -5

!? HashMap 간단 설명

- HashMap은 Key, Value를 저장하는 Map의 구현체 중 하나입니다.
- 자료구조에 Key를 넣으면 Value를 반환하도록 합니다.
- HashMap은 Key를 Hashing 하여 저장하고 빠르게 처리하여 HashMap이란 입력과 삭제에 대해 시간복잡도가 $O(1)$ 입니다.

다음과 같은 데이터가 주어진다는 가정을 했을 때
("원천동", "대기오염업체")
("원천동", "수질오염업체")
("원천동", "대기오염업체")
("원천동", "마트")
("원천동", "마트")
("원천동", "편의점")

2중 HashMap을 사용하면 다음과 '원천동'이라는 key를 가지는 2중 HashMap으로 데이터를 정리할 수 있다.

```
{  
  원천동 = {  
    대기오염업체 = 2,  
    수질오염업체 = 1,  
    마트 = 2,  
    편의점 = 1  
  }  
}
```

장단점

- 장점 : Open Addressing이 아닌 Seperate Chaining 방식을 사용하기 때문에 추가, 삭제가 용이하며 동기화를 지원하지 않기 때문에 처리속도가 빠르다.
- 단점 : 현재 사용하지 않는 Key에 대해서 미리 Memory를 할당받아 사용하고 있는 단점이 있다.

4. 전체 검색창 (KMP Algorithm)

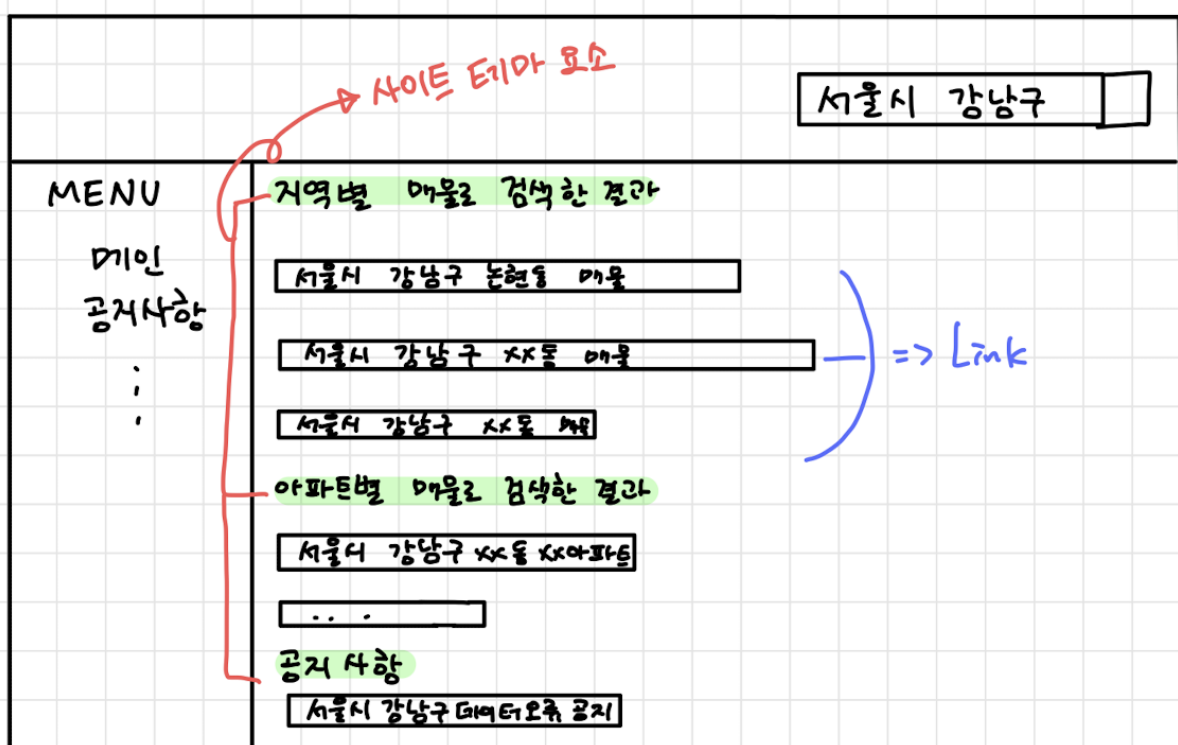
💬 적용 부분

1. 메인 화면 상단 검색 Bar

🖋 구현 목적

- 홈페이지의 모든 정보를 검색하여 해당 키워드가 포함된 데이터들의 Link를 결과페이지의 List로 보여줌
- 검색 키워드로 해당하는 데이터 정보들을 전부 뽑아오고 기준 별로 각 테마에 데이터를 뿌림

→ 기준은 각 테마의 DB 테이블 별로 나누어 지역별 테마에는 region 테이블에서 해당 키워드로 찾은 정보를 뿌리고, 아파트별 테마에는 brand 테이블에서 키워드로 검색한 정보를 뿌리고, 공지사항 테마는 notice 테이블에서 조회 후 데이터를 뿌려주는 방식으로 이용



KMP 알고리즘 간단 설명

- KMP 알고리즘의 시간 복잡도는 $O(N+M)$ 으로 빠른 속도를 가짐
- KMP 알고리즘은 탐색시 매 위치마다 새로 탐색을 시작하는 것이 아닌 이전 탐색의 결과를 재사용

장단점

- 장점 : KMP 알고리즘은 불일치가 일어났을 때 다음으로 비교할 위치를 빨리 찾아내고 이전 탐색의 결과를 재사용하므로 효율적인 탐색 가능
- 단점 : 알파벳의 수가 커지면 Brute Force 알고리즘과 비슷하게 비효율적인 탐색이 일어난다.

5. 매물 추천 기능 (CBF 알고리즘) - 관심 지역

적용 부분

1. 찜 매물과 유사도가 높은 매물을 추천해주는 기능

구현 목적

- 사용자가 찜한 매물과 유사도가 높은 매물을 추천해줌으로써 사용자에게 더 유용하고 많은 정보 제공을 위함

!? CBF 간단 설명

- CBF는 Content Based Filder의 약자로, 사용자가 읽은 데이터와 다른 데이터들의 유사도를 측정하여 제공하는 추천 알고리즘이다.

장단점

- 장점 : 새로운 상품을 추천하지 못하는 First-Rate 문제를 해결할 수 있다는 장점이 있다.
- 단점 : 새로운 장르의 추천이 어렵다.

거래 정보 차트화 기능 (Quick Sort 알고리즘)

적용 부분

1. 메인화면 하단의 지역별 평균 매매가 순위 차트

구현 목적

- 각 지역의 매매가 평균을 구한 뒤 해당 평균값을 정렬하여 높은 평균가를 가지는 순서대로 출력



! Quick Sort 알고리즘 간단 설명

- 불안정 정렬에 속하며, 다른 원소와의 비교만으로 정렬을 수행하는 비교 정렬
- 분할 정복 알고리즘의 하나로, 평균적으로 매우 빠른 수행 속도를 자랑
- 병합 정렬과 달리 퀵 정렬은 리스트를 비균등하게 분할 함

장단점

- 장점 : 시간 복잡도가 $O(N \times \log N)$ 을 가지는 다른 정렬 알고리즘과 비교했을 때도 가장 빠르다.
병합정렬과 다르게 추가 메모리 공간을 필요로 하지 않는다.
- 단점 : 정렬된 리스트에 대해서는 퀵정렬의 불균형 분할에 의해 오히려 수행시간이 더 많이 걸린다.