lv1.데이터분석

-sort함수의 커스터마이징

sort 함수를 커스터마이징 할 때에는 sort(vec.begin(), vec.end(), ~~~~) 세번째에 존재하는 ~~~~부분에 내가 원하는 조건을 넣으면 된다. 마지막에 return해야 하는 부분을 넣어야 한다. 마지막 부분에 들어가는 비교함수는 bool을 반환해야 한다.//

비교함수의 원칙 : 첫번째 인자가 두번째 인자보다 앞에 와야 하면 true, 그렇지 않으면 false가 와야 한다.

-map의 특성과 사용

map은 map<key type, value type, compare> 이름; 이런 식으로 정의 되는데 만약 compare에 해당하는 값이 없다면 알아서 key로 오름차순 정렬된다.

-map과 pair

map이랑 pair는 사실 둘이 같은거라고 볼 수 있다. pair에 map의 고유 특성 몇가지를 더하면 map이 된다. 그래서

삽입시에 map.insert(pair)🡪자료형이 pair이어야 한다!!

-unordered map 헤더파일

map이지만 정렬이 따로 되지 않고,,name to idx….

-remove if, erase if 사용법

vector 컨테이너를 다룰 때 remove\_if와 erase\_if를 사용하게 되는 경우가 발생한다.

먼저 remove\_if는 ‘a’,’b’,’c’,’d’가 벡터 안에 존재할 때, 만약 remove\_if b를 한다면 b가 맨 마지막으로 가게 정렬 된 후에 다시 처음을 가리키는 iterator를 반환하는 함수이고

erase\_if의 경우 요소들을 삭제하고 재정렬 해준다. 이건 c++20에서부터 가능하기 때문에 remove\_if로 뒤로 가게 한 후 erase를 통해 따로 지워주는 작업이 필요하다.

cpp를 보면 초록 색 글씨로 remove if 와 erase를 설명해 놓았다

-data.end()는 맨 마지막 요소가 아니라 맨 마지막보다 한 칸 뒤에있는 논리적 끝을 가리키는 반복자를 반환

그래서 data.end()를 출력하려고 하거나 참조하면 오류가 발생하고 data.end()-1로 참조해야 한다. 반면에 data.begin()은 첫번째 요소를 가리키는 것이 맞다.

-Lambda 람다식 사용

람다식을 사용하면 익명의 함수를 빠르게 정의 하지 않고 사용할 수 있다는 장점이 있다.

먼저 람다식의 문법을 살펴보면

[introducer capture] (parameter) -> return type {statement} 이렇게 이루어져 있다…함수 객체에 대한 이해가 사라짐,, 함수 객체란..?