

그룹 애너그램

문제 : 문자열 배열을 받아 애너그램 단위로 그룹핑하라.

예시1)

Input: strs = ["eat","tea","tan","ate","nat","bat"]

Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

예시2)

Input: strs = [""]

Output: ""

예시3)

Input: strs = ["a"]

Output: "a"

애너그램을 판단하는 가장 간단한 방법은 정렬하여 비교하는 것이다. 애너그램을 이루는 단어들을 정렬하면 모두 같은 값을 갖게 되기 때문이다. 우선, 정렬한 값을 보관하기 위해 맵을 선언한다. 하나의 키에 여러개의 값이 들어갈 수 있기 때문에 값은 리스트 형태가 되어야 한다.

```
Map<String, List<String>> results = new HashMap<>();
```

입력값을 하나씩 순회하면서 정렬한다.

```
for (String s : strs){
    char[] chars = s.toCharArray();
    Arrays.sort(chars);

    String key = String.valueOf(chars);

    if(!results.containsKey(key)){
        results.put(key, new ArrayList<>());
    }

    results.get(key).add(s);
}
```

```
[
    ["ate", "eat", "tea"],
    ["ant", "tan"],
```

```
        ["cat"]  
    ]
```

리트코드 문제에서 정답 문자열 배열을 다시 `List<List<String>>` 을 리턴하도록 되어있다.
이 경우는 키가 필요없는 형태이기 때문에 값만 추출해서 반환해주면 된다.

```
return new ArrayList<>(results.values());
```

```
public List<List<String>> groupAnagrams(String[] strs) {  
    Map<String, List<String>> results = new HashMap<>();  
  
    for (String s : strs) {  
        // 문자열을 문자 배열로 변환  
        char[] chars = s.toCharArray();  
        Arrays.sort(chars);  
        // 문자 배열을 키로 하기 위해 다시 문자열로 변환  
        String key = String.valueOf(chars);  
  
        // 만약 기존에 없던 키라면 빈 리스트를 삽입  
        if (!results.containsKey(key))  
            results.put(key, new ArrayList<>());  
        // 키에 해당하는 리스트에 추가  
        results.get(key).add(s);  
    }  
    return new ArrayList<>(results.values());  
}
```