

Report

프로그래머스 가장 큰 수 문제 (42746)



과목 : 코딩테스트지도

담당교수 : 이승진 교수님

학부 : IT융합자율학부

학번 : 202114136

이름 : 장준희

제출일 : 2025-03-20

작업 git url: <https://github.com/jjune960/coding-assignments/>

문제 설명

0 또는 양의 정수가 주어졌을 때, 정수를 이어 붙여 만들 수 있는 가장 큰 수를 알아내주세요.

예를 들어, 주어진 정수가 [6, 10, 2]라면 [6102, 6210, 1062, 1026, 2610, 2106]를 만들 수 있고, 이중 가장 큰 수는 6210입니다.

0 또는 양의 정수가 담긴 배열 numbers가 매개변수로 주어질 때, 순서를 재배치하여 만들 수 있는 가장 큰 수를 문자열로 바꾸어 return 하도록 solution 함수를 작성해주세요.

제한 사항

- numbers의 길이는 1 이상 100,000 이하입니다.
- numbers의 원소는 0 이상 1,000 이하입니다.
- 정답이 너무 클 수 있으니 문자열로 바꾸어 return 합니다.

요구사항 정리

주어진 배열 numbers에서 순서를 재배치하여 가장 큰 수를 문자열로 return하는 문제입니다.

1차 시도

numbers의 순서를 재배치하는 것을 생각해 보았다.

일반적으로 numbers 재배치는 for문을 사용해서 한다.

for문 안에 무엇이 들어가면 좋을까?

Numbers 배열에서 number을 불러온 뒤에 해야 할 것(for문 내부):

number의 첫번째 자리수를 구한다.

기존 숫자의 최대 숫자보다 첫번째 자릿수가 큰지 비교한다.

만약 같다면 두번째 자릿수를 비교한다. (이하 반복)

큰 자릿수를 찾았다면 그 숫자를 numbers의 첫번째 배열의 숫자와 맞바꾼다.

이를 모든 numbers 배열에 반복하면 주어진 배열을 재배치하여 가장 큰 숫자를 얻을 수 있다!

이제 가장 큰 자릿수의 숫자를 구해야 한다.

10으로 나누고 몫이 0인 때의 나머지를 구한다.

1차 구현 (스스로 시도)

1. 자릿수 구하기

먼저 자릿수를 구해보겠다.

원본 숫자를 보존하기 위해 임시변수를 만들었다. (tempnum)

tempnum이 0이 될 때까지 반복하고 카운트를 한다. (0부터 카운트)

코드:

```
while (tempnum / 10 != 0) { // 자릿수 구하기
    tempnum = tempnum / 10;
    count++;
}
```

12345 숫자로 테스트 결과 4 (0부터 카운트를 시작했으므로 정상적인 결과)

2. 첫 자릿수 추출하기

카운트한 개수의 바로 직전 까지만 계산하면 첫 자릿수가 나온다.

코드:

```
tempnum = number; // 임시 변수 다시 사용
for(int i = 0; i < count; i++) { // 카운트한 개수의 바로 직전 까지만 계산하면
    tempnum = tempnum / 10;
}
```

12345 숫자로 테스트 결과 1

이 방법으로 시도했으나 가장 큰 자릿수만 나오고 두번째 자릿수부터는 다양한 경우의 수를 고려하려니 난이도가 크게 증가하였다.

2차 시도 (인터넷, AI 등 참고)

숫자의 자릿수를 하나씩 추출하는 방식은 구현이 복잡하다.

대신 숫자 2개를 문자열로 취급해 이어 붙이고 그것에 대한 크기를 비교하는 방법을 사용하자.

1. int 배열을 문자열 배열로 변환하기

숫자를 이어붙이려면 문자열 형태로 변환하는 것이 적합하다.

코드:

```
String[] numStrs = new String[numbers.length];
for (int i = 0; i < numbers.length; i++) {
    numStrs[i] = String.valueOf(numbers[i]);
}
```

2. 내림차순으로 정렬(숫자 큰 순서)하기

내림차순으로 두 숫자를 내림차순으로 비교한 후 정렬한다.

Comparator<String> 익명 클래스를 구현한다.

코드:

```
Arrays.sort(numStrs, new Comparator<String>() {
    public int compare(String s1, String s2) {
        return (s2 + s1).compareTo(s1 + s2);
    }
});
```

3. 0인 경우 처리하기

가장 큰 숫자가 0이면 모두 0이므로 answer 값으로 0을 반환한다.

가장 큰 숫자는 index 0에 존재한다.

코드:

```
if (numStrs[0].equals("0")) { // 가장 큰 숫자가 0인 경우
    answer = "0";
}
```

4. 문자열 이어 붙여 결과 생성하기

for 문으로 문자열을 이어붙여 answer에 넣는다.

코드:

```
else {
    for (String num : numStrs) { // 문자열 이어 붙이기
        answer += num;
    }
}
```

이 방식은 각 숫자의 자릿수를 개별적으로 비교하는 복잡한 로직을 피하면서, 두 숫자를 이어 붙였을 때의 전체 값을 정확히 반영한다.

비교 기준 (s2+s1).compareTo(s1+s2) 덕분에, 어떤 숫자 조합이 최종 결과에서 우위를 가지는지 일관성 있게 정할 수 있다.

배운 점

처음에는 각 숫자의 자릿수를 개별적으로 비교하려고 했지만,

- **자릿수 비교의 복잡성:**

각 숫자의 첫 번째 자리수가 같을 경우 이후 자리수를 비교하는 로직을 구현하기가 매우 복잡해졌습니다.

- **예외 상황 처리 어려움:**

숫자마다 자릿수의 개수가 다르기 때문에, 일관되게 비교 기준을 정하기가 어려웠습니다.

이러한 문제를 해결하기 위해,

문자열을 붙여서 비교하는 방식을 도입하였습니다.

- 두 숫자를 문자열로 변환한 후, "A+B"와 "B+A"의 결과를 비교하면,
두 숫자를 이어 붙였을 때 전체 숫자의 크기를 직접적으로 반영할 수 있습니다.

따라서, 이 방법을 이용하면 주어진 배열 numbers에서 숫자들의 순서를 재배치하여,
이어 붙였을 때 가장 큰 수를 문자열로 반환할 수 있습니다.

git commit hash

Commits on Mar 22, 2025

docs: 가장 큰 수 문제의 배운 점 문서로 정리 & pdf 제작 (2차 시도 성공 및 전체 일부 수정) jjune960 committed 1 minute ago	9aff865		
feat: 가장 큰 수 2차 시도 (성공) jjune960 committed 8 minutes ago	dba1ad0		
feat: 가장 큰 수 2차 시도 (4. 문자열 이어 붙여 결과 생성하기) jjune960 committed 42 minutes ago	4b614d8		
feat: 가장 큰 수 2차 시도 (3. 0인 경우 처리하기) jjune960 committed 1 hour ago	b59743e		
feat: 가장 큰 수 2차 시도 (2. 내림차순으로 정렬(숫자 큰 순서)하기) jjune960 committed 1 hour ago	97d2da9		
feat: 가장 큰 수 2차 시도 (1. int 배열을 문자열 배열로 변환하기) jjune960 committed 1 hour ago	97b26b1		

Commits on Mar 21, 2025

feat: 가장 큰 수 1차 시도 (실패) jjune960 committed 14 hours ago	90799ca		
feat: 가장 큰 수 1차 시도 (2. 첫 자릿수 추출하기) jjune960 committed 16 hours ago	a26d1eb		
feat: 가장 큰 수 1차 시도 (1. 자릿수 구하기) jjune960 committed 16 hours ago	3a50d16		
docs: 가장 큰 수 문서로 정리 (1차 시도 가설) jjune960 committed 18 hours ago	fd7922f		