

2331번: 반복수열

2331번 제출 맞은 사람 슛코딩 풀이 풀이 작성 풀이 요청 재채점/수정 문제 추천

채점 현황 내 소스 강의▼ 질문 검색 질문 작성

반복수열 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	2792	1167	889	41.484%

문제

다음과 같이 정의된 수열이 있다.

- $D[1] = A$
- $D[n] = D[n-1]$ 의 각 자리의 숫자를 P 번 곱한 수들의 합

예를 들어 $A=57$, $P=2$ 일 때, 수열 D 는 $\{57, 74(=5^2+7^2=25+49), 65, 61, 37, 58, 89, 145, 42, 20, 4, 16, 37, \dots\}$ 이 된다. 그 뒤에는 앞서 나온 수들(57부터가 아니라 58부터)이 반복된다.

이와 같은 수열을 계속 구하다 보면 언젠가 이와 같은 반복수열이 된다. 이 때, 반복되는 부분을 제외했을 때, 수열에 남게 되는 수들의 개수를 구하는 프로그램을 작성하시오. 위의 예에서는 $\{57, 74, 65, 61\}$ 의 네 개의 수가 남게 된다.

문제 유형 : 자료구조, 자료 탐색, 중복 되는 수 찾기, 중복되는 순간 찾기

접근 방법 :

문제 유형에 그래프라고 되어 있어서 들어오는 숫자를 그래프의 노드라고 생각하였다. 그 후 문제에 주어진 규칙에 의해 다음 숫자를 생성 후 그 숫자를 다음 노드로 보았다. 이런 식으로 계속 진행 하면, 방문 한 노드로 이어지는 순간이 발생하고 그 노드가 루프의 시작이다. 그래프로 구현 하려면 노드와 노드가 가진 데이터는 노드가 만들어진 순서(Integer : i ($i \geq 1$))이고, 이를 이용해 구현하면 된다.

그런데 굳이 그래프를 위해 인접 행렬이나, 인접 리스트를 사용하고 dfs같은 탐색 알고리즘을 구현할 필요가 없어 보였다. dict유형 자료 구조

예를 들어 파이썬의 딕셔너리를 이용해 key는 노드의 숫자와 value는 노드가 만들어진 순서를 업데이트한다. 같은 노드가 만들어지는 순간 딕셔너리의 메소드를 이용해 확인 후 그 루프 노드의 시작이 몇 번 째인지 안 후 그 보다 1 작은 숫자를 출력하면 된다.

python

```
n , p = [int(i) for i in input().split()]
```

```
def cal(x):
```

```
    s = 0
```

```
    while(x) :
```

```
        s+= int(pow(x%10,p))
```

```
        x/=10
```

```
    return s
```

```
d = dict()
```

```
d.update({n:1})
```

```
ans = 0
```

```
while 1 :
```

```
    prv = n
```

```
    n = cal(n)
```

```
    if d.__contains__(n) :
```

```
        ans = d[n] -1
```

```
        print(ans)
```

```
        break
```

```
    else :
```

```
        d.setdefault(n,d[prv] +1)
```

java

```
import java.io.BufferedReader;
```

```
import java.io.InputStreamReader;
```

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
public class Main {
```

```
    public static Map<Integer,Integer> m = new HashMap<>();
```

```
    public static BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
```

```
    public static void main(String args[])throws Exception {
```

```
        String[] line = in.readLine().split(" ");
```

```
        int n = Integer.parseInt(line[0]), p =Integer.parseInt(line[1]);
```

```
        m.put(n,1);
```

```
        while(true) {
```

```
            int cur = n;
```

```
            n = cal(cur,p);
```

```
            if(m.containsKey(n)) {
```

```
                System.out.println(m.get(n)-1);
```

```
                break;
```

```

        } else {
            m.put(n, m.get(cur)+1);
        }
    }
}

public static int cal(int cur,int p) {
    int c = cur;
    int sum =0;
    while (c>0) {
        sum += (int)Math.pow((c%10),p);
        c/=10;
    }
    return sum;
}
}

```

입력

첫째 줄에 $A(1 \leq A \leq 9999)$, $P(1 \leq P \leq 5)$ 가 주어진다.

출력

첫째 줄에 반복되는 부분을 제외했을 때, 수열에 남게 되는 수들의 개수를 출력한다.

예제 입력

57 2

예제 출력

4

힌트