



# 백준 1256: 사전

[통계](#) [수정](#) [삭제](#)

zeong · 방금 전

0

[c언어](#) [백준](#) [사전](#) [알고리즘](#)

## 알고리즘스터디

▼ [목록 보기](#)

8/8



## 백준 1256 사전

### [문제 요약]

특별한 문자열로 이루어진 사전 만들기  
사전에 수록되어 있는 모든 문자열은 N개의 "a"와 M개의 "z"로 이루어져 있고 다른 문자는 없음.  
사전에는 알파벳 순서대로 수록되어 있음.  
N과 M이 주어졌을 때, 사전에서 K번째 문자열이 무엇인지 구하기.

### [주의 사항]

1. 사전에 수록되어 있는 문자열의 개수가 K보다 작으면 -1을 출력한다.
2. 시간 제한은 2초이다.
3. 문자열을 만들어야 한다.

### [문제 해결 과정]

보통 우리가 생각하는 사전은 abc, 가나다 등 단어를 찾는 것이기 때문에, 문자열을 입력받는다.  
하지만 이 문제에서는 입력받는 것이 문자열이 아닌 숫자이기 때문에, az 사전에 수록되어 있는 문자열을 대략적으로 알고 시작해야할 것 같았다.



$N = 1, M = 3 \rightarrow azzz / zazz / zzaz / zzza$

$N = 2, M = 1 \rightarrow aaz / aza / zaa$

$N = 2, M = 2 \rightarrow aazz / azaz / azza / zaaz / zaza / zzaa$

az 사전이 만들어지는 원리는 대충 알 것 같기도 한데, 코드로 구현하려니까 막막하기만 하다.

"N과 M이 주어졌을 때, 사전에서 K번째 문자열이 무엇인지 구하는 프로그램을 작성하시오."를 구현하기는 쉬운데, 문자열 만드는데 복병이다..

사전 ADT 알고리즘 로직을 보고 문제를 풀고 있었는데, 아무리 생각해도 이걸 이용해서 문제를 풀 수 없을 것만 같았다.

위에 내가 만들어 놓은 az사전을 이용해서 처음부터 다시 시작했다.

문제를 풀다가 이 문제를 과연 몇 명이나 풀었을까 궁금해서 찾아봤는데, c언어로 풀었을 때 맞힌 사람이 56명밖에 없다는게 싸했다.

도저히 못 풀겠어서 gpt에게 도와달라고 했다.

주석을 달면서 코드를 해석해보는데, 솔직히 잘 이해가 안 된다.

gpt가 도와준 코드도 시간초과가 나긴 했는데, 값은 그럴싸하게 나온다.

너무 어렵다..

이 문제를 풀 수 있으면 정말 코딩 고수가 될 것 같은 느낌이다.

하지만 지금 내 실력으로는 풀 수 없을 것 같다..

## [개념]

문제를 풀기 전에 찾아본건데, 이 문제를 풀 때는 필요없는 것 같다..

사전 ① ② ③ ④

사전내용과 비슷해 보이는 문제?? ① ② ③

## [ gpt 코드 - 시간초과 ]

```
#include <stdio.h>
#include<string.h>
#include<stdlib.h>
#pragma warning(disable:4996)

char result[201]; //문자열 result 생성

long long binomialCoeff(int n, int k) { // 이항 계수를 계산하는 함수
    if (k == 0 || k == n) //k가 0이거나 n과 같으면
    {
        return 1; //1을 return한다
    }
    return binomialCoeff(n - 1, k - 1) + binomialCoeff(n - 1, k);
}
```



```

if (K > binomialCoeff(N + M, N)) {
    printf("-1"); // K가 가능한 문자열의 수보다 크면 -1 출력
    return;
}

while (N > 0 && M > 0) { //N과 M이 모두 0보다 클 때
    long long combination = binomialCoeff(N + M - 1, N - 1);

    if (K <= combination) {
        result[index++] = 'a'; //문자열에 a 대입
        N--; //N값 1 감소
    }
    else {
        result[index++] = 'z'; //문자열에 z 대입
        M--; //M값 1 감소
        K -= combination; //K에 (K - combination) 값 대입
    }
}

while (N > 0) { //N이 0보다 클 때
    result[index++] = 'a'; //문자열에 a 대입
    N--; //N값 1 감소
}

while (M > 0) { //M이 0보다 클 때
    result[index++] = 'z'; //문자열에 z 대입
    M--; //M값 1 감소
}

result[index] = '\0'; //NULL값 대입
printf("%s\n", result); //문자열 출력
}

int main() {
    int N, M;
    long long K;

    scanf("%d %d %lld", &N, &M, &K);

    find(N, M, K);

    return 0;
}

```



효정



## 0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by  
**Stellate**