



Week5_과제

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 8분 전 · 비공개

0

[1256:사전](#)

Smarcle_23Algorithm



▼ 목록 보기

6/6



Baekjoon_1256 : 사전

순서대로 문자열을 생성하여 각 인덱스에 저장할 것이기 때문에 무순사전 사용

풀이 과정

1. 입력받은 n, m으로 사전순 문자열들을 생성
2. 구조체의 key에 순서 인덱스를, element에 생성된 문자열을 저장
3. k번째 key에 저장된 element를 출력

문제 해결

- 아무것도 출력이 안됨
동적할당 된 list 크기를 키워보니 출력됨
- 런타임 에러 n, m이 크면 출력이 안됨
동적할당의 크기가 너무 크면 오래걸리고 작으면 출력이 안됨-> 비트시프트 연산을 사용하여 $1 \ll (n+m)$ 크기로 할당 : $(2^{(n+m)})$ 를 의미 but, k가 크면 출력이 안됨
- 메모리 초과
구조체를 지우고 current에 문자열을 저장하여 k번째일때 출력하는 것으로 변경-> n과 m이 클때는 출력이 되지만 k가 클때는 시간이 너무 오래걸림

코드

내 코드

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int count = 1, cnt = 0;

void generateStrings(char* current, int n, int m, int nCount, int mCount, int index, int k) {
    if (cnt == 1) return;

    if (index == n + m) {
        current[index] = '\\0';
        if (count == k) {
            printf("%s", current);
            cnt = 1;
            return;
        }
        count++;
    }

    if (nCount < n) {
        current[index] = 'a';
        generateStrings(current, n, m, nCount + 1, mCount, index + 1, k);
    }

    if (mCount < m) {
        current[index] = 'z';
        generateStrings(current, n, m, nCount, mCount + 1, index + 1, k);
    }
}

int main() {
    int n, m, k;
    scanf("%d %d %d", &n, &m, &k);

    char* current = (char*)malloc((n + m + 1) * sizeof(char));

    generateStrings(current, n, m, 0, 0, 0, k);

    if (cnt == 0) printf("-1");

    free(current);

    return 0;
}

```

무순사전X, 실행은 되지만 시간이 오래걸림 -> **실패**

gpt 코드

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

unsigned long long binomialCoeff(int n, int k) {
    if (k == 0 || k == n) {
        return 1;
    }
}

```

```

    unsigned long long res = 1;
    k = k < (n - k) ? k : (n - k);
    for (int i = 0; i < k; ++i) {
        res *= (n - i);
        res /= (i + 1);
    }
    return res;
}

void generateKthString(char* current, int n, int m, unsigned long long k) {
    if (n == 0 && m == 0) {
        current[0] = '\0';
        return;
    }

    if (n > 0) {
        unsigned long long possibilities = binomialCoeff(n + m - 1, n - 1);
        if (k <= possibilities) {
            current[0] = 'a';
            generateKthString(current + 1, n - 1, m, k);
            return;
        }
        else {
            current[0] = 'z';
            generateKthString(current + 1, n, m - 1, k - possibilities);
            return;
        }
    }
    else {
        current[0] = 'z';
        generateKthString(current + 1, n, m - 1, k);
        return;
    }
}

int main() {
    int n, m;
    unsigned long long k;
    scanf("%d %d %llu", &n, &m, &k);

    if (n < 1 || n > 100 || m < 1 || m > 100 || k < 1 || k > binomialCoeff(n + m, n)) {
        printf("-1");
        return 0;
    }

    char* result = (char*)malloc((n + m + 1) * sizeof(char));
    generateKthString(result, n, m, k);

    if (result[0] == '\0') {
        printf("-1");
    }
    else {
        printf("%s", result);
    }

    free(result);

    return 0;
}

```

이항계수를 사용하는 이유와 unsigned long long에 대한 이해 부족



유정



이전 포스트
Week4_과제

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate