[백준] 사전

통계 수정 삭제

imhyun · 방금 전

0

백준 사전 알고리즘

https://www.acmicpc.net/problem/1256

> N개의 'a'와 M개의 'z'로 이뤄진 사전이 있다. 사전은 알파벳 순서로 되어 있고 그럴 때 K번째의 단어를 찾는 문제이다.

아이디어

나의 처음 발상은 아래와 같았다.

> 1. **사전을 만든다.**
>   - N개의 'a'와 M개의 'z'로 이뤄진 단어를 만든다.
>   - 랜덤하게 만들어 사전 배열에 넣는다.
> 2. **K번째 문자열을 찾는다**
>   - 사전의 단어수를 먼저 파악해 k와 비교.
>   - 조건이 맞다면 정렬해 단어를 찾는다.

이에 따라 단어를 랜덤하게 만드는 함수를 아래와 같이 만들었다.

단어 랜덤 형성 함수

```
Alg makeWord
    input N,M number of a,z
    output word consists of a,z

    1.word[N + M]<-'\0'
     i<-0
    2.while(N>0&M>0)
```

```
            if(a%2=0)
                N<-N-1
                word[i]<-'a'
            else
                M<-M-1
                word[i]<-'z'

            i<-i+1
        3.while(N>0)
            word[i]<-'a'
            i<-i+1
            N<-N-1
        4.while(M>0)
            word[i]<-'z'
            i<-i+1
            M<-M-1
```

실제 코드

```
char* makeWord2(int N, int M) {

    char* word = (char*)malloc(sizeof(char) * (N + M + 1));

    word[N + M] = '\0';
    int i = 0;

    srand(time(NULL));

    while ((N > 0) && (M > 0)) {

        if (rand() % 2 == 0) {
            N--;
            word[i] = 'a';
        }

        else {
            M--;
            word[i] = 'z';
        }

        i++;
    }

    while (N > 0) {
        word[i] = 'a';
        i++;
        N--;
    }

    while (M > 0) {
        word[i] = 'z';
        i++;
        M--;
    }

    return word;
}
```

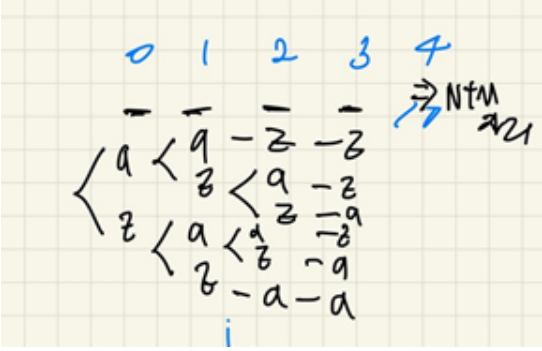
단어 랜덤 형성 함수의 문제점

~~하지만 치명적인 오류가 있었으니, 계속해서 돌려도 rand()는 같은 값이 나오기 때문에 내가 원하는 것처럼 랜덤하게 단어가 형성되지 않다는 것이다.~~ **(이는 난수를 생성하는 시드값을 설정하는 함수 srand를 사용하지 않았기 때문이다.)**

또한 랜덤하게 단어가 형성되면 같은 것은 저장하지 않도록 같은 것을 계속 찾아야 하고, **사전 단어 개수에 맞게 채워질 때까지 지속적으로 기다리게** 될 수 있다는 단점이 있었다.

따라서 **재귀 형식**으로 처리해야 했다.

재귀 형식 단어 형성 함수



예시로 주어진 N=2,M=2를 먼저 살펴보았을 때 위에 그림과 같이 나타나는 것을 알 수 있다. 나는 매 과정이 **N,M이 0보다 크다면 둘 중 하나를 선택**, 하나가 0이라면 **무조건 존재하는 단어를 배치**해주면 된다는 규칙을 찾았다.

또한 4자리의 단어를 만든다는 것은 맨 앞에 1자리 단어를 위에처럼 선택해주고 3자리 단어를 채운다는 것과 같은 의미였다. 즉 재귀 형태가 이뤄지는 것이다.

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

void makeWord(char*,int,int, int,int,int*);

int main(void)
{
    //=======================================정보 입력=============================================
    int N, M, K;

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

    char* word = (char*)malloc(sizeof(char) * (N + M + 1));

    int cnt = 0;
    //=====================================사전 단어 개수=========================================
    unsigned long long size=1;

    if (N == 0||M==0)
```

```
            size = 1;

        else {
            for (int i = N + M; i >= 1; i--)
                size *= i;

            for (int i = N ; i >= 1; i--)
                size /= i;

            for (int i = M; i >= 1; i--)
                size /= i;

        }

        //================================사전 단어 개수 판별================================
        if (size < K) {
            printf("-1\n");
        }

        else {
            word[N + M] = '\0';
            makeWord(word, 0, N, M,K,&cnt);
        }
        /*char** Dict;

        char* word = (char*)malloc(sizeof(char) * (N + M+1));

        makeWord(word,0,N, M);

        printf("%s",*Dict[K]);*/
    }

    void makeWord(char* word,int i, int N, int M,int K,int *cnt) {

        if (N <= 0) {
            while (M > 0) {
                word[i++] = 'z';
                M--;
            }
        }

        else if (M <= 0) {
            while (N > 0) {
                word[i] = 'a';
                i++;
                N--;
            }
        }

        else if ((N <= 0) && (M <= 0)) {
        }

        else {
            word[i] = 'a';
            makeWord(word,i + 1,N-1,M,K,cnt);

            word[i] = 'z';
            makeWord(word,i + 1, N, M-1,K,cnt);

        }

        if ((N <= 0) && (M <= 0)) {
```

```
            (*cnt)++;
            if (*cnt == K)
                printf("%s\n", word);
        }

    }
```

박시현

이전 포스트
[백준] 힙과 힙정렬

0개의 댓글

댓글을 작성하세요

댓글 작성