

433. Minimum Genetic Mutation

☰ Tags	DFS medium stack
📅 Property	@November 2, 2022
🔗 URL	

Question

原文：

A gene string can be represented by an 8-character long string, with choices from 'A', 'C', 'G', and 'T'.

Suppose we need to investigate a mutation from a gene string `start` to a gene string `end` where one mutation is defined as one single character changed in the gene string.

- For example, "AACCGGTT" --> "AACCGGTA" is one mutation.

There is also a gene bank `bank` that records all the valid gene mutations. A gene must be in `bank` to make it a valid gene string.

Given the two gene strings `start` and `end` and the gene bank `bank`, return *the minimum number of mutations needed to mutate from `start` to `end`*. If there is no such a mutation, return `-1`.

Note that the starting point is assumed to be valid, so it might not be included in the bank.

我的理解：

給各一段8個字元的基因，分別是初始基因，以及突變後的基因，和基因庫bank，基因一次只會改變一個位置，改變後的基因也只有基因庫已經有的樣式，如果沒有辦法透過基因庫一步一步改變成突變後的基因就返回return -1，如果可以的話return 最少的突變次數

翻譯：

自評翻譯正確性：

- Word Memory :
 - mutation 突變

Code

```
class Solution {
public:
    int minMutation(string start, string end, vector<string>& bank) {
        //stepCounter 用來記錄所有可能步數
        //step步數
        vector<int>stepCounter;
```

```

int step=0;
int i=0;

int test=0;
// dfs 會蒐集各個路線的步數
dfs(start,end,bank,stepCounter,step);

//透過 dfs 蒐集完各個路線的步數回傳先排序
sort(stepCounter.begin(),stepCounter.end());
int ans=100;

//排序後第一個出現不是 -1 的數就是最少步數
//如果全部都是 -1 表示 start 無法轉變到 end
for( i=0;i<stepCounter.size();i++ ){
    if((stepCounter[i]!=-1) && (ans > stepCounter[i])){
        ans=stepCounter[i];
        break;
    }
    else if(i==stepCounter.size()-1){
        ans=-1;
    }
}
return ans;
}

//傳進來的有起始基因、目標基因、基因庫、收集大家步數的 vector、計步器
void dfs(string start, string end, vector<string>bank , vector<int>& stepCounter,int step){

    int i=0,j=0;
    int counterror=0;
    stack<string>st;
    string temp="";

    step++;
    //先把起始基因從基因庫抹掉，避免往回走
    for( i=0;i<bank.size();i++ ){
        if(bank[i]==start){
            bank.erase(bank.begin()+i);
        }
    }

    //比對起始基因跟基因庫的基因
    //如果兩者只差一個位元進判斷
    //如果這個可以轉化的基因庫基因等於目標基因，等於這次轉變就可以轉化成功，於是將本路線步數放進 stepCounter
    //不等於的話先放進 stack
    for( i=0;i<bank.size();i++ ){
        counterror=0;
        for( j=0;j<8;j++ ){
            if( start[j]!=bank[i][j] ){
                counterror++;
            }
        }
        if(counterror==1){
            if(bank[i]==end){
                stepCounter.push_back(step);
            }
            else{
                st.push(bank[i]);
            }
        }
    }

    //將 stack 中的元素放進 dfs
    for( j=0;j<st.size();j++ ){
        dfs(st.top(),end,bank,stepCounter,step);
        st.pop();
    }
}

```

```

    }

    //因為一開始有先從 bank 裡 erase掉 start，所以bank的元素會越來越少
    //出現跟當前start剛好差一字元的元素機率也降低，如果start沒有辦法轉變到end且也沒有剛好差一字元的元素存在bank
    //表示該路線已到盡頭，就將 -1 放進步數統計vector
    if(st.empty()){
        stepCounter.push_back(-1);
    }

    }
};

```

思路：透過DFS的思路，遍歷所有路線，將各路線結果記錄下來，如果可以轉變成end紀錄步數，如果無法轉變過去紀錄-1，最後在排序有記錄所有路線結果的vector，並掃過一次如果有-1以外的數出現(第一個)，那個數就是最短路徑，如果全部結果都是 -1 表示所有路線都失敗 start 不能轉變至 end

優良code參考

我時間 0 牛逼轟轟 就不參考別人了(被揍

思路：