第4次编程作业 $Report$

罗华坤 软件02 2019011799

1 重名剔除

思路

由于餐品个数有限，选取哈希数为600001，尽可能地减少重复的可能。

为了防止编码大小超过int，采取字母所处位置 * 字母表的序号数的编码方式。

每次添加菜名，检查时候存在重复或者已经重复过便可。

实现方法

首先先对输入编码，按照上述的编码方式，具体如下：

```
long long int encode(char *c){
    long long int code = 0;
    int len = strlen(c);
    for(int i = 0; i < len; i++){
        code += (c[i] - 'a' + 1) * (i + 1);
    }
    return code;
}
```

接着查看对应桶内是否存在，若存在重复则输出；若已重复，直接退出；若不重复则插入桶中。

2 游戏

思路

根据上课的提示，采用康托展开将全排列编码成连续的数，储存在一个长度固定的数组内。

在运行一开始，先将所有种可能连接形成连接图，在查询时采用BFS遍历图直至查询到为止。

实现方法

采用一长度固定数组储存不同状态以及可以变化得到的下一状态，以及是否被遍历过的标志。

然后在运行之前，从初始状态开始遍历全图，设所有状态数为 n，则构建的复杂度为$O(n)$

查询时从该状态采用BFS直至到达初始状态，若全图均遍历则失败。

时间复杂度

由于边的个数与状态数相关且大致成线性关系，因此BFS的时间复杂度$O(|V|+|E|)=O(n)$

设查找次数为m，则总的时间复杂度为
$T(n)=T_{预处理}(n)+T_{查找}(n)=O(n)+O(mn)=O(mn)$

3 任务调度

思路

写一个优先级队列（小顶堆）便可，包括上滤、下滤操作，最后按顺序提取便可，思路较为简单。

实现方法

问题主要在提取时，需要将提取出来的数乘2放回该优先级队列，还需判断与上限的关系。

提取时还需要注意优先级内是否为空以避免一些特殊情况。

```
struct node{//节点
    char* content;//内容
    long long int pri;//优先级
    bool operator<(const node* other){...}//重载运算符
}
struct Priority{//优先级队列
    node Node[MAXN];
    int size = 0;
    int parent(int i){return (i - 1) >> 1;}
    int lChild(int i){return (i << 1) + 1;}
    int rChild(int i){return lChild(i) + 1;}
    void upFlow(int i){//上滤
        while(i > 0 && Node[i] < Node[parent(i)]){
            swap(node[i], node[parent(i)]);
            i = parent(i);
        }
    }
    void downFlow(int i){...}
    void insert(node a){//插入
        Node[size ++] = a;
        upFlow(size - 1);
    }
    node extract(){//唯一特殊之处
        node min = Node[0];
        node[0] = node[--size];
        downFlow(0);
        if(min.pri * 2 < MAX){
            min.pri *= 2;
            insert(min);
        }
        return min;
    }
}
```