

华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：21 级

上机实践成绩：

指导教师：金澈清

姓名：杨茜雅

上机实践名称：动态表

学号：

上机实践日期：5.12

10215501435

上机实践编号：No.11

一、目的

1. 熟悉算法设计的基本思想
2. 熟悉动态表的插入和删除

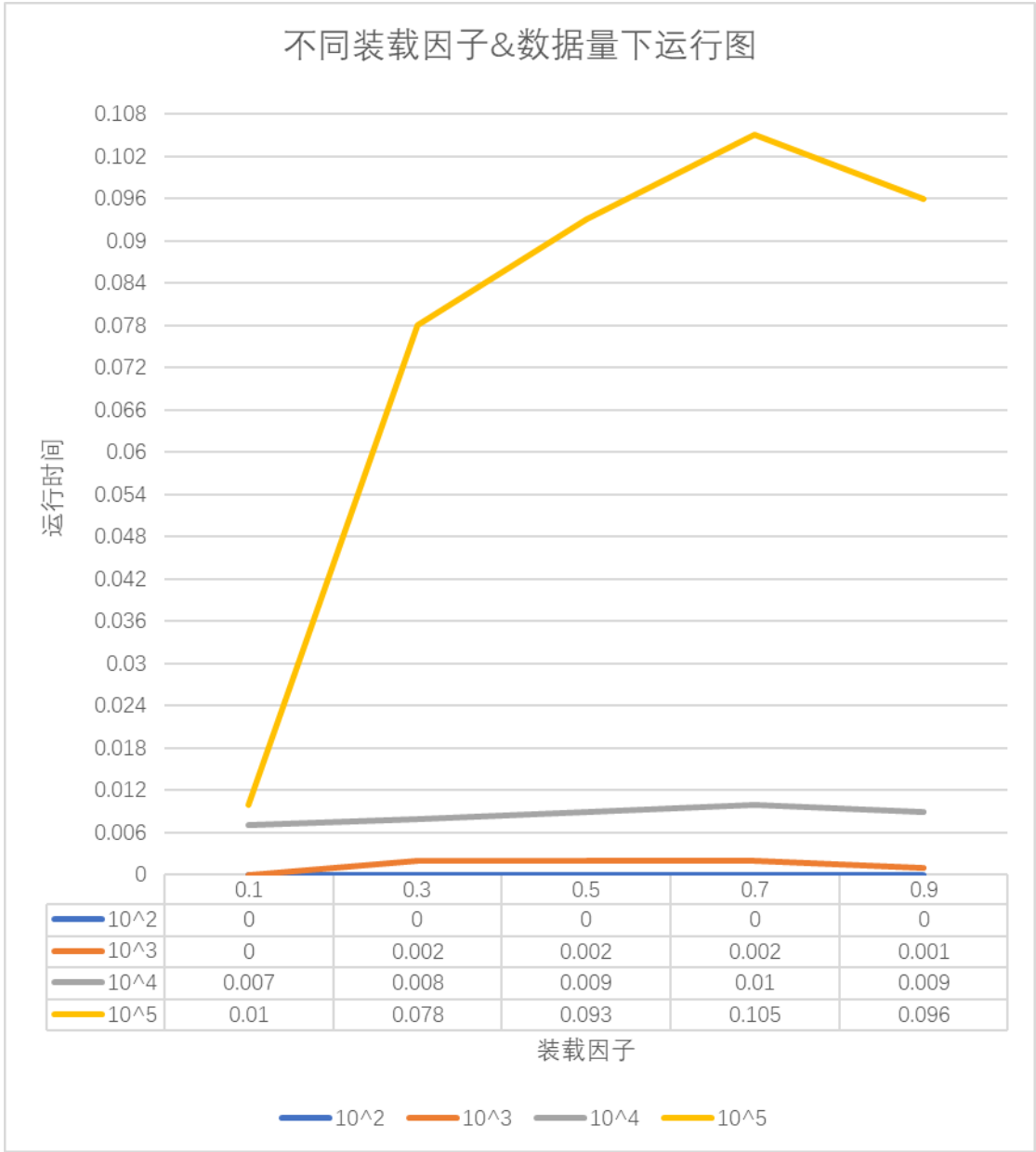
二、内容与设计思想

1. 小明前不久开发一款学生信息管理系统，并打算给学校们试用，每个学校都有 10G 的额度。系统中存储的是学生的学号（11 位，仅有数字）&姓名（10 位，仅有字母）信息。但是显然，不是每个学校都会用满所有的额度，绝大多数只会用到一小部分。于是为了提高系统的利用率，小明不会在一开始就给每个学校开辟一块空的 10G 的空间，而是根据学校存储的数据量进行弹性伸缩。请你帮小明设计一个可以动态调整大小的动态表，均支持插入&删除操作。
2. 用不同规模的操作次数（ $10^2, 10^3, 10^4, 10^5$ ）和不同的装载因子（0.1, 0.3, 0.5, 0.7, 0.9）进行实验，分析不同情况下动态表的性能。

三、使用环境

推荐使用 C/C++集成编译环境。

四、实验过程



五、总结

```
#include<bits/stdc++.h>
#include<cmath>
#include <time.h>
#include<string.h>
#define factor 0.7//装载因子
#define First_size 10//初始表长度
using namespace std;

class Table
{
```

```
public:
    Table();
    void Table_Print();
    void Table_Insert(string &student_number,string
&student_name);
    void Table_Delete();
    void Table_expandspace();//扩张
    void Table_shrinkspace();//收缩
    int Empty();//判断是不是空的
    {
        if(item==0)
            return 1;
        else
            return 0;
    }
private:
    int item;//表中元素个数
    int maxsize;//最多可以放几个
    string *total_number;
    string *total_name;
};

Table::Table();//初始化
{
    total_number=new string[First_size];
    total_name=new string[First_size];
    item=0;
    maxsize=First_size;
    for(int i=0;i<maxsize;i++)
    {
        total_number[i]="\0";
        total_name[i]="\0";
    }
}
```

```
void Table::Table_Print()
{
    for(int i=0;i<item;i++)
    {
        cout<<total_number[i]<<' '<<total_name[i]<<endl;
    }
}

void Table::Table_expandspace()
{
    string *a=total_number;
    string *b=total_name;
    total_number=new string[2*maxsize];//扩张到两倍
    total_name=new string[2*maxsize];
    for(int i=0;i<item;i++)
    {
        total_number[i]=a[i];//移过去
        total_name[i]=b[i];
    }
    maxsize=2*maxsize;
    delete[]a;
    delete[]b;
}

void Table::Table_shrinkspace()
{
    string *a=total_number;
    string *b=total_name;
    int c=ceil(maxsize*0.5);//向上取整
    total_number=new string[c];
    int d=ceil(maxsize*0.5);
    total_name=new string[d];
    for(int i=0;i<item;i++)
    {
```

```
        total_number[i]=a[i];
        total_name[i]=b[i];
    }
    delete[]a;
    delete[]b;
    maxsize=maxsize*factor;//表收缩
}

void Table::Table_Insert(string &student_number,string
&student_name)
{
    if(item>=maxsize)
    {
        Table_expandspace();
    }
    total_name[item]=student_name;
    total_number[item]=student_number;
    item++; //插入后元素个数记得加一
}

void Table:: Table_Delete()
{
    total_name[item-1]="\0";
    total_number[item-1]="\0";
    item--;
    if(item<factor*maxsize)
    {
        Table_shrinkspace();
    }
}

string rand_name(int length)//random 生成 10 位字母
{
    string str;
```

```
char c;
for(int i=1;i<=length;i++)
{
    c='a'+rand()%26;
    str.push_back(c);
}
return str;
}

string rand_number(int length)//random 生成 11 位学号
{
    string str;
    char c;
    for(int i=1;i<=length;i++)
    {
        c='0'+rand()%10;
        str.push_back(c);
    }
    return str;
}

int main()
{
    srand((unsigned)time(NULL));
    Table Table;
    string name;
    string number;
    int N;
    cin>>N;
    int a=1+rand()%2;//a 随机取 1 或者 2，1 则删除，2 则插入，以此模拟插入删除随机来
    double start=clock();
    for(int i=1;i<=N;i++)
    {
```

```
    if(a==1)
    {
        if(Table.Empty()==0)
        {
            Table.Table_Delete();
        }

    }
    else if(a==2)
    {
        name=rand_name(10);
        number=rand_number(11);
        Table.Table_Insert(number,name);
    }
}
double end=clock();
double diff=(double)(end-start)/CLOCKS_PER_SEC;
cout<<endl<<diff;
    return 0;
}
```