

数据结构 2022秋 Lab3

TA: 张皓捷 19302010021

说明

本Lab主要关于特殊矩阵和KMP算法的内容

任务

1. 还原对称矩阵

我们知道 $n * n$ 的对称矩阵可以用长度为 $\frac{n(n+1)}{2}$ 的一维数组表示。

例如我们有矩阵

$$A = \begin{Bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{Bmatrix} \quad (1)$$

其中

$$a_{i,j} = a_{j,i} \quad (2)$$

则我们可以用数组

$$V = \{a_{1,1} \quad a_{2,1} \quad a_{2,2} \quad a_{3,1} \quad a_{3,2} \quad a_{3,3} \quad \dots \quad a_{n,1} \quad a_{n,2} \quad \dots \quad a_{n,n}\} \quad (3)$$

来表示矩阵 A ，即只保存 A 中下三角的所有元素。

现给你数组 V ，请你还原出矩阵 A 。

例如传入 $compressedMatrix = \{1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6\}$ ，则应该返回 $\begin{Bmatrix} 1 & 2 & 4 \\ 2 & 3 & 5 \\ 4 & 5 & 6 \end{Bmatrix}$ ，以

`vector<vector<int>>`（二维vector）的形式返回。

```
vector<vector<int>> > SpecialMatrixOps::recoverSymmetricMatrix(const vector<int>
&compressedMatrix) {
    //TODO
}
```

2. 实现KMP算法

请根据课上讲的内容，使用KMP算法实现`strStr`函数。

`strStr`函数在字符串`haystack`中查找字符串`needle`的第一次出现下标。如果`haystack`中找不到`needle`，则返回-1。字符串下标是zero-based的。

例如`strStr("abcdabc", "bcd")`返回1；而`strStr("abcdabc", "bcde")`返回-1。

```
int KMP::strStr(const string &haystack, const string &needle) {  
    //TODO  
}
```

请在getNextArr中实现计算needle的next数组的过程。

```
std::vector<int> KMP::getNextArr(const string &needle) {  
    //TODO  
}
```

提示

- 你需要修改的文件有且仅有src/KMP.cpp和src/SpecialMatrixOps.cpp。请不要修改其他文件。
- 请确保你的代码能正常编译，并通过tests/KMPTest.cpp和tests/SpecialMatrixOpsTest.cpp中的所有测试用例后再提交。如遇到困难可联系ZHJ。
- 任务2中，我提供了一个对于暴力(BF)算法来说极坏的测试用例。对于该测试用例，BF算法需要数十秒才能完成计算，而使用KMP算法只需要数百毫秒。我设置了最大2000毫秒的运行时间限制，因此使用BF算法无法通过测试。你可以利用这个机会，比较KMP算法和BF算法的效率区别。

截止日期

2022年10月16日 23:59

提交

将src/KMP.cpp和src/SpecialMatrixOps.cpp打包成zip，上传到elearning。

文件命名为 学号-姓名-Lab3.zip，例如21302019999-花卷-Lab3.zip。