



数据结构 & 算法



GLIMMER-RECRUITMENT

数据结构小实践

好久没做招新题了，是不是已经觉得寂寞了？我也好久没有出题了，那就两全其美，我们来布置一次小任务来充实一下大家的技术学习罢（

本题出题：YEWPO 以及YEWPO强烈要求指明题面是tk_sky写的

今天tk_sky又在窥屏算法大手子YEWPO，看看他在学什么。只见他屏幕上显示了一个名字非常好听的数据结构：**Dancing List**。会跳舞的链表，就冲这个名字，tk_sky决定他不管怎么样也要学会它。

趁tk_sky正在搜索，你也来学习一下吧。那就从最简单的二维数组开始好了。

本题的数据：

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 100 \end{bmatrix}$$

Part 1 二维数组

Task 1 创建

用 `c` 声明上面的二维数组吧（方式任意）。

Task 2 查找

在这个数组里查找67。（我知道你知道67在哪）。

你设计的查找方式时间效率高吗？（该方法的时间复杂度如何？）

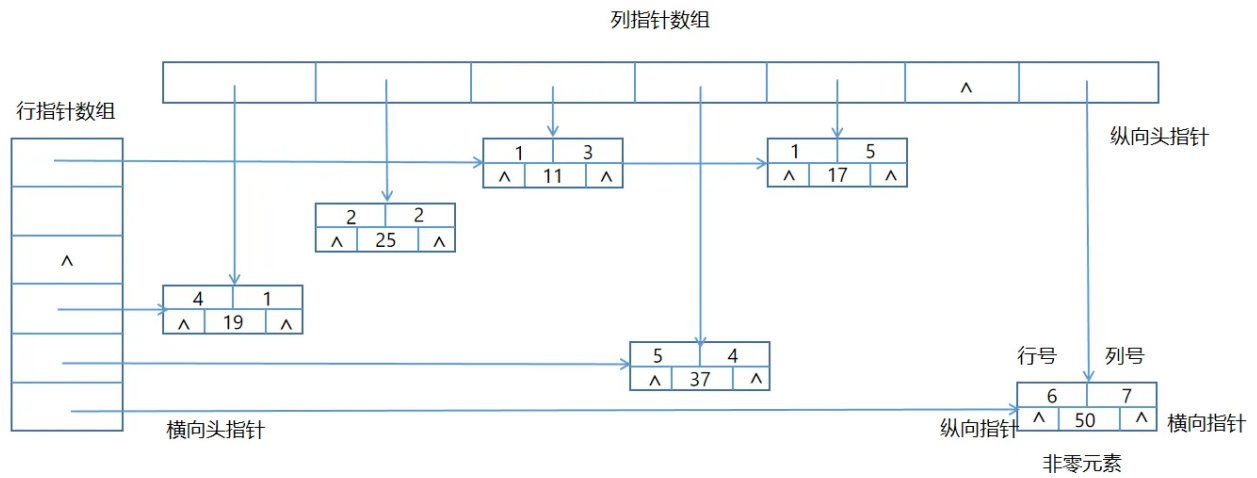
Part 2 dancing list

等你写完这个二维数组，tk_sky貌似还在等他的搜索结果。那不如直接进入主题吧。

你发现，你写的这个二维数组中数值0占了90%，有实际值的数据只占了10%，我们称这种矩阵为稀疏矩阵。十字链表（dancing list）的出现解决了在稀疏矩阵中查找效率低的问题。

十字链表中，每个非零元素都可以在他所在的行链表和纵链表中找到，即确定行号和列号，就可以准确的确定元素的位置。

十字链表的例图如下：



https://blog.csdn.net/qg_27870421

Task 3 写一个会跳舞的链表吧

现在你已经是DancingList大师了。请你实现这个可爱的数据结构吧。

该数据结构需实现以下功能：

- 数据结构初始化
- 插入元素
- 查找元素
- 删除元素

写好后将题目上的矩阵储存在你的数据结构中，并查找67，并回答此时的时间复杂度如何？你觉得十字链表可以运用在那些地方？

尾声

tk_sky貌似还在等他的搜索结果读条。难道今天工作室的wifi坏了吗？真是奇怪。

