

report

week 8

程雨歌 12307110079

2016 年 4 月 23 日

1 findValue.h

1.1 函数说明

在稀疏矩阵中找到相应坐标的元素

输入 一个指向待查找的稀疏矩阵指针 *t，两个整型 i,j 分别表示待查找的行和列。

输出 一个元素类型，即在该稀疏矩阵 T 中相应坐标上所找到的元素 T[i][j]。若未找到则返回 0。

实现方法 依次扫描稀疏矩阵的元素，利用元素关于先行后列的字典序排列的性质，先跳到行号不小于所查找行位置的第一个元素，然后继续向后查找行号列号均相等的位置，返回其元素值。期间若有任何闪失均返回 0。

1.2 复杂度

时间复杂度 $\mathcal{O}(n)$ n 为稀疏矩阵元素个数

空间复杂度 $\mathcal{O}(1)$

1.3 边界情况

- 传入空稀疏矩阵指针将得到 0；
- 查找行列号为负数或超过指定稀疏矩阵的行列数将得到 0；
- 注：所查找位置为从 0 开始的下标。

1.4 程序运行结果

```
Your Matrix C:
Row: 3, Col: 4, nums: 1
(1,2): 3

Expected:
0 0 0 0
0 0 3 0
0 0 0 0
Press Enter key to continue...
```

2 multiMat.h

2.1 函数说明

计算两个稀疏矩阵的乘积并以稀疏矩阵形式返回。

输入 指向两个稀疏矩阵的指针 *matA, *matB。

输出 指向运算结果的稀疏矩阵指针 *matC。

实现方法 在输入矩阵可以相乘的情况下，声明一个正常的矩阵 C，在进行完之后的累加以后将 C 转换为稀疏矩阵输出。

只有 A（第一个矩阵）中列号与 B（第二个矩阵）中行号相等的元素才会被乘到一起再累加到相应位置。依次扫描 A 中每个元素，对每个元素 a_{ij} 将能乘到一起的 B 中元素 b_{jk} 找到后相乘再累加到矩阵 C 的相应位置 c_{ik} 。

但对于 a_{ij} 若对每一个 $0 \leq k < \text{matB} \rightarrow \text{colSize}$ 都用前面的 findValue 函数查找一下 b_{jk} ，复杂度将达到 $\mathcal{O}(N_A N_B Z)$ ，其中 N_A, N_B 分别为矩阵 A 和 B 中元素个数，Z 为 B 的列数。

所以我们先对 B 进行预处理，利用元素行号单调递增的特性，将每一行第一个元素在元素列表中的位置标记出来 $\text{bH}[r] = n_r$ ， $0 \leq r \leq Y$ ，Y 为 B 的行数 A 的列数。若某行没有元素则仍标记为其后第一个元素位置，也就是每一行存储着行号不大于它的第一个元素的位置。对于不存在 Y+1 行即 $\text{bH}[Y]$ 则存储元素总数。这样所有上下界形式一致了，B 中第 j 行的所有元素 b 在元素列表中位置 n 满足 $\text{bH}[j] \leq n < \text{bH}[j+1]$ ，其中 $0 \leq j < Y$ 。

进行完预处理后，可以对于每个 a_{ij} 直接找到所有可乘的 b_{jk} ，将时间复杂度降到 $\mathcal{O}(N_A N_B)$

2.2 复杂度

时间复杂度 $\mathcal{O}(N_A N_B + Y + XZ)$ ，X 为 A 的行数

$N_A N_B$ 为计算乘积时间，Y 为预处理 B 所需时间，XZ 为将 C 转换为稀疏矩阵存储形式时间。

空间复杂度 $\mathcal{O}(Y + XZ)$

2.3 边界情况

- 若有输入的稀疏矩阵指针为 NULL，则返回 NULL；
- 若两个矩阵不能相乘，则返回 NULL；
- 若有矩阵的行或列不为正，则返回 NULL；
- 若乘积结果是零矩阵，由于在声明矩阵 C 时使用 calloc 已经全部初始化为 0 元素了，结果正确。

2.4 程序运行结果

```
(4,0): 2
(4,1): 3
Press Enter key to continue...

Your Matrix C:
    null

Expected: null
Press Enter key to continue...
```