

不同的图的格式之间的转换 tsD14409

计 72 陈嘉杰

March 12, 2018

Contents

1	题目说明	1
2	实现思路	1
3	程序编译环境	2
4	实现步骤	2
4.1	下标约定	2
4.2	数据的读入	2
4.3	输出关联矩阵	3
4.4	输出边列表	3
4.5	输出正向表	4
4.6	输出邻接表	5
4.7	完整代码	5
5	遇到的问题和得到的收获	7

1 题目说明

文件 `input.txt` 中含有一个图的权矩阵表示，要求输出这个图的关联矩阵、边列表、正向表和邻接表到 `output.txt` 中。

2 实现思路

先把传入的这个权矩阵转化为边的列表和邻接表。边列表用于关联矩阵和边列表的转换，邻接表可以用于正向表的生成。

3 程序编译环境

1. 操作系统: macOS
2. 编译器: LLVM/Clang 6.0.0

4 实现步骤

4.1 下标约定

输入数据中点和边都从 1 开始, 故在我的代码中同样如此。部分与点和边无关的下标, 则从 0 开始。

4.2 数据的读入

考虑到输入的数据中 $0 < n \leq 100$, 所以可以直接在全局变量中开足够大的空间存放数据。首先是文件重定向, 接着, 读入数据, 转换成边的列表和邻接表的格式。

```
1 #include <stdio.h>
2
3 int edge_from[10010];
4 int edge_to[10010];
5 int edge_weight[10010];
6 int edge_num = 1;
7 int adj[110][10010] = {{0}};
8 int adj_weight[110][10010] = {{0}};
9 int adj_num[110] = {0};
10
11 int main() {
12     freopen("input.txt", "r", stdin);
13     freopen("output.txt", "w", stdout);
14     scanf("%d", &n);
15     for (int i = 1; i <= n; i++) {
16         for (int j = 1; j <= n; j++) {
17             int weight;
18             scanf("%d", &weight);
19             if (weight) {
20                 edge_from[edge_num] = i;
21                 edge_to[edge_num] = j;
22                 edge_weight[edge_num] = weight;
```

```

23         edge_num++;
24
25         adj[i][adj_num[i]] = j;
26         adj_weight[i][adj_num[i]] = weight;
27         adj_num[i] ++;
28     }
29 }
30 }
31 return 0;
32 }

```

4.3 输出关联矩阵

关联矩阵中每一列对应一条边，只要把这条边的两个结点所在的行进行处理即可。处理完则输出。

```

1 // guanlian
2 for (int i = 1; i < edge_num; i++) {
3     link_matrix[edge_from[i]][i] = 1;
4     link_matrix[edge_to[i]][i] = -1;
5 }
6 for (int i = 1; i <= n; i++) {
7     for (int j = 1; j < edge_num; j++) {
8         printf("%d ", link_matrix[i][j]);
9     }
10    printf("\n");
11 }

```

4.4 输出边列表

边列表，在读入数据的时候已经处理好，简单处理成矩阵的形式，直接输出即可。

```

1 // bianliebiao
2 for (int i = 1; i < edge_num; i++) {
3     edge_list[0][i] = edge_from[i];
4     edge_list[1][i] = edge_to[i];
5     edge_list[2][i] = edge_weight[i];
6 }
7
8 for (int i = 0; i < 3; i++) {

```

```

9   for (int j = 1; j < edge_num; j++) {
10      printf("%d ", edge_list[i][j]);
11   }
12   printf("\n");
13 }

```

4.5 输出正向表

这个可能就是最难得一部分了。这个表中，A 数组的元素代表相应的结点的后继结点的起始下标。最后添加一个结尾。所以，我们使用变量 zhengxiangbiao_current 记录当前的下标，以此更新 A 数组。

```

1  // zhengxiangbiao
2  for (int i = 1; i <= n; i++) {
3      zhengxiangbiao_a[i] = zhengxiangbiao_current;
4      for (int j = 0; j < adj_num[i]; j++) {
5          zhengxiangbiao_b[zhengxiangbiao_current] = adj[i][j];
6          zhengxiangbiao_z[zhengxiangbiao_current] =
            ↪ adj_weight[i][j];
7          zhengxiangbiao_current ++;
8      }
9  }
10 zhengxiangbiao_a[n+1] = zhengxiangbiao_current;
11
12 for (int i = 1; i <= n+1; i++) {
13     printf("%d ", zhengxiangbiao_a[i]);
14 }
15 printf("\n");
16
17 for (int i = 1; i < zhengxiangbiao_current; i++) {
18     printf("%d ", zhengxiangbiao_b[i]);
19 }
20 printf("\n");
21
22 for (int i = 1; i < zhengxiangbiao_current; i++) {
23     printf("%d ", zhengxiangbiao_z[i]);
24 }
25 printf("\n");

```

4.6 输出邻接表

这一步也很简单，在读入数据的时候已经处理完毕。

```
1 // linjiebiao
2 for (int i = 1; i <= n; i++) {
3     for (int j = 0; j < adj_num[i]; j++) {
4         printf("%d %d ", adj_weight[i][j], adj[i][j]);
5     }
6     printf("\n");
7 }
```

4.7 完整代码

```
1 #include <stdio.h>
2
3 int edge_from[10010];
4 int edge_to[10010];
5 int edge_weight[10010];
6 int edge_num = 1;
7 int adj[110][10010] = {{0}};
8 int adj_weight[110][10010] = {{0}};
9 int adj_num[110] = {0};
10 int link_matrix[110][10010] = {{0}};
11 int edge_list[3][10010] = {0};
12 int zhengxiangbiao_a[110] = {0};
13 int zhengxiangbiao_b[10010] = {0};
14 int zhengxiangbiao_z[10010] = {0};
15 int zhengxiangbiao_current = 1;
16 int n;
17
18 int main() {
19     freopen("input.txt", "r", stdin);
20     freopen("output.txt", "w", stdout);
21     scanf("%d", &n);
22     for (int i = 1; i <= n; i++) {
23         for (int j = 1; j <= n; j++) {
24             int weight;
25             scanf("%d", &weight);
26             if (weight) {
27                 edge_from[edge_num] = i;
```

```

28         edge_to[edge_num] = j;
29         edge_weight[edge_num] = weight;
30         edge_num++;
31
32         adj[i][adj_num[i]] = j;
33         adj_weight[i][adj_num[i]] = weight;
34         adj_num[i] ++;
35     }
36 }
37 }
38
39 // guanlian
40 for (int i = 1; i < edge_num; i++) {
41     link_matrix[edge_from[i]][i] = 1;
42     link_matrix[edge_to[i]][i] = -1;
43 }
44 for (int i = 1; i <= n; i++) {
45     for (int j = 1; j < edge_num; j++) {
46         printf("%d ", link_matrix[i][j]);
47     }
48     printf("\n");
49 }
50
51 // bianliebiao
52 for (int i = 1; i < edge_num; i++) {
53     edge_list[0][i] = edge_from[i];
54     edge_list[1][i] = edge_to[i];
55     edge_list[2][i] = edge_weight[i];
56 }
57
58 for (int i = 0; i < 3; i++) {
59     for (int j = 1; j < edge_num; j++) {
60         printf("%d ", edge_list[i][j]);
61     }
62     printf("\n");
63 }
64
65 // zhengxiangbiao
66 for (int i = 1; i <= n; i++) {
67     zhengxiangbiao_a[i] = zhengxiangbiao_current;

```

```

68     for (int j = 0; j < adj_num[i]; j++) {
69         zhengxiangbiao_b[zhengxiangbiao_current] = adj[i][j];
70         zhengxiangbiao_z[zhengxiangbiao_current] =
            ↪ adj_weight[i][j];
71         zhengxiangbiao_current ++;
72     }
73 }
74 zhengxiangbiao_a[n+1] = zhengxiangbiao_current;
75
76 for (int i = 1; i <= n+1; i++) {
77     printf("%d ", zhengxiangbiao_a[i]);
78 }
79 printf("\n");
80
81 for (int i = 1; i < zhengxiangbiao_current; i++) {
82     printf("%d ", zhengxiangbiao_b[i]);
83 }
84 printf("\n");
85
86 for (int i = 1; i < zhengxiangbiao_current; i++) {
87     printf("%d ", zhengxiangbiao_z[i]);
88 }
89 printf("\n");
90
91 // linjiebiao
92 for (int i = 1; i <= n; i++) {
93     for (int j = 0; j < adj_num[i]; j++) {
94         printf("%d %d ", adj_weight[i][j], adj[i][j]);
95     }
96     printf("\n");
97 }
98 return 0;
99 }

```

5 遇到的问题和得到的收获

遇到的问题主要就是审题不仔细，忘记输出权值，因此有了两次的错误提交。改掉之后就成功 AC 了。得到的收获就是，即便是写一个很简单的程序，最好也要提前想好思路和容易写错的地方随时提醒自己。这样可以省出

更多的时间，省出时间来完成其它的作业。