Problem Set 8

麻超　201300066

南京大学人工智能学院

目录

1 Problem 1

第 1-2 行, 分别建立 x_1 到 x_16 的集合.

第 3-4 行之后, 为:$\{x_1,x_2\},\{x_3,x_4\},...,\{x_{15},x_{16}\}$

第 5-6 行之后:$\{x_1,x_2,x_3,x_4\},...,\{x_{13},x_{14},x_{15},x_{16}\}$

第 7-8 行之后:$\{x_1,x_2,x_3,\ldots,x_8\}\{x_9,x_{10},x_{11},\ldots,x_{16}\}$

第 9 行之后:$\{x_1,x_2,x_3,x_4,...x_{14},x_{15},x_{16}\}$ 所以两次 find 操作返回的值都是 x_1.

2 Problem 2

首先做 n 次 $MakeSet(x_i)$ 操作, 操作完成之后可以获得 n 个元素个数为 1 的集合.

再两两做 $Union$ 操作, 可以获得 $\lceil\frac{n}{2}\rceil$ 个有两个元素的集合... 依次类推, 仿照第一题的操作, 可以得到一个高为 $\lg n+1$ 的 n 元集合, 一共需要做 $2^{\lfloor \lg n \rfloor}-1$ 次 Union() 操作.

最后再进行 m-n+1 次 Find 操作. 知道 Find 操作作用在了之前构建的高度为 $\lg n+1$ 的树上, 因为 $m \geq 2n$ 所以其时间复杂度为 $\Omega(m\lg n)$.

3 Problem 3

3.1 a

最坏情况下, 所有的元素都存在在同一棵树的同一侧, 即形成了一条链, 链的长度为 n, 所需时间复杂度 $T(n)=\sum_{i=0}^{n} i=\frac{n(n+1)}{2}$, 即 $T(n)=O(n^2)$

3.2 b

用路径压缩实施 Find 操作, 假设在该过程中进行了 i 次路径压缩, 总时间复杂度 $T(n)=\sum_{k=1}^{i} m_k+(n-i)=O(n).m_k$ 为每次路径压缩的时间复杂度,(n-i) 为其余 n-i 个元素所需要的时间复杂度.

即使用路径压缩完成 Find 操作后, 总时间复杂度为 $O(n)$.

4 Problem 4

5 Problem 5

利用数学归纳法证明:

1. 若这个图只有两个节点, 则这个树两个节点中间有且只有一个边相连, 即两个节点均只有一度.

2. 假设该连通无环图共有 k 个节点, 且该 k 个节点中至少有两个点的度数为 1, 记为 A 和 B.

3. 则考虑该连通无环图从 k 个节点增加至 k+1 个节点的情况:

- 若增加的该节点与 A 和 B 之间没有连通的边, 那么无论其他的边如何变化, 无论它究竟还是不是连通无环图, 这两个节点 A 和 B 自然都是 1 度的, 所以其正确.

- 若增加的节点与 A 有一条边连通 (与 B 也是同理), 但与 B 没有连通, 那么增加的节点 S 必然不会与原图中其他节点连通, 因为原图是连通的, 所以若 S 与其他节点连通且与 A 连通, 那么自然会形成一个回路, 就违背了无环的前提. 在这种情况下,B 的度数为 1,A 虽然与 S 连通, 度数变为了 2, 但 S 不与其他节点连通, 其度数同样为 1, 则证明正确.

- 若 S 与 A 和 B 都连通, 由于原图是联通的, 所以 A 和 B 自然有一条通路, 这样连通相当于构造了一个环, 违背了前提.

综上所述, 原命题正确, 即任何具有 $n \geq 2$ 个顶点的连通无环图至少有两个顶点的度数为 1.

6 Problem 6

假设 $A = BB^T$, 对 A 中的元素 a_{ij} 分情况讨论:

第一种情况, 当 $i = j$ 时, 就是用 B 的第 i 行去乘 B^T 的第 j 列 (也就是 B 的第 j 行), 此时

$$a_{ij} = \sum_{k=1}^{|E|} b_{ik}^2$$

即为对 B 的第 i 行元素求平方和, 只有当节点 i 与第 k 条边相连时方可取 1, 则 a_{ij} 表示与节点 i 相连的边数.

第二种情况, 当 $i \neq j$ 时, 用 B 的第 i 行乘以 B 的第 j 行

$$a_{ij} = \sum_{k=1}^{|E|} b_{ik} \cdot b_{jk}$$

而 $b_{ik} \cdot b_{jk}$ 的值只能是-1 或 0, 当 i 和 j 节点间存在一条边时取-1, 否则取 0, 则 a_{ij} 表示与节点 i 相连的边数.

综上所述,BB^T 中的每一个元素 a_{ij} 都表示节点 i 与节点 j 相连的边数.