1. 并查集知识点总结：
2. 每次Union操作，不同集合的数目至少减1，在同一集合的两个元素合并不会导致集合数目减少；
3. Find操作的效率与树的高度相关，树的高度有三种优化方法：link-by-size、link-by-rank、path compression；
4. 根节点一旦变成非根节点就不会再变成根节点了；
5. Path compression操作之后，树的高度可能降低，但节点的rank值不变；
6. 上机中的蝴蝶分类问题本质上是带偏移量并查集的特殊情况，偏移量只有0和1。POJ上的食物链问题是带偏移量并查集的经典应用，Code22；
7. 对于判断两点是否连通的问题，如果一开始给定图，则。用DFS，如果边是依次增加，则使用并查集。
8. Percolation问题中并查集的使用

上机题目中已经有一道Percolation的问题，解决方法是在首行和末行加入超级节点，并默认打开且可以和下/上层任意节点相连。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  | 1 |  |  |  |  |
|  | 2 | 3 |  |  |  |
|  |  | 7 | 4 |  |  |
|  |  | 5 |  |  |  |
|  |  | 6 | 8 |  |  |
|  | 9 |  | 10 |  |  |
|  | | | | | |

变式1：如果问从上灌水，最多能灌多少个格子的水？

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 |  |  | 1 |  |
|  | 1 | 1 |  | 1 | 1 |
|  |  | 1 | 1 |  |  |
|  |  |  | 1 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

如图，本题答案为9，一种解法是在首行上面添加超级节点即可，之后判断超级节点所在集合元素的个数，减去超级节点数，即为答案。

变式2：对于多种问题的组合，可以考虑使用多个图同时操作，比如一个图上下都有超级节点，另外一个只有上面有超级节点。

变式3：如果打开的格子还能关上？

不能使用并查集，因为并查集不支持分割操作。