

CS2302 - Data Structures

Spring 2019

Lab # 6

Disjoint Set Forests

Deadline: Friday, April 12, 2019

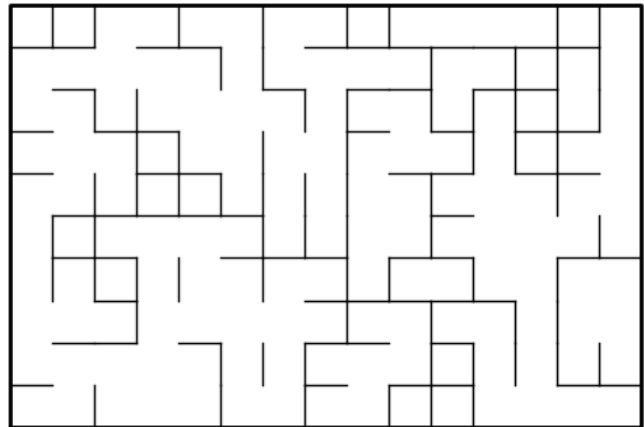
In this assignment you will use a disjoint set forest to build a maze. Your maze should contain a collection of cells separated by walls in such a way that there is exactly one simple path (that is, a path that does not visit any cell more than once) separating any two cells.

To build a maze, you must do the following. Let M be the number of rows and N be the number of columns of your square maze. When all walls are present (see figure), each of the $M * N$ cells in the maze belongs to a different set. Thus you have $M * N$ sets in your disjoint set forest. When you remove a wall, if the cells that were separated by that wall belonged to different sets, you must unite these sets. This process is repeated until all cells belong to a single set; at that point you display the maze. The following pseudocode illustrates the process to build the maze:

```
Create full maze with all adjacent cells are separated by a wall
Assign each cell to a different set in a disjoint set forest S
While S has more than one set
    Select a random wall w =[c1,c2]
    If cells c1 and c2 belong to different sets, remove w and join c1's set and c2's set
    otherwise do nothing
Display maze
```

| | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 |
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
| 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 |
| 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

Initial maze, including cell numbers



Maze after removing randomly-chosen walls

The program *maze.py*, provided in the class web page, contains code for all drawing operations and to remove randomly chosen walls from the maze. You will need to modify the program to remove the walls that will result in a correct maze.

Perform a comparison of running times of the program using standard union and union by size with path compression for various maze sizes. As usual, write a report describing your work.