Lab 7
Maze

2019. 04. 18

lab 7. Maze

Example Step

Pick (8,14)

Start

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

End

S
{1,2,7,8,9,13,19}
{3}
{4}
{5}
{6}
{10}
{11,17}
{12}
{14,20,26,27}
{15,16,21}
.
.
{22,23,24,29,30,32
33,34,35,36}

21

Example

S
{1,2,7,8,9,13,19}
{3}
{4}
{5}
{6}
{10}
{11,17}
{12}
{14,20,26,27}
{15,16,21}
.
.
{22,23,24,29,39,32
33,34,35,36}

Find(8) = 7
Find(14) = 20

Union(7,20)

S
{1,2,7,8,9,13,19,14,20 26,27}
{3}
{4}
{5}
{6}
{10}
{11,17}
{12}
{15,16,21}
.
.
{22,23,24,29,39,32
33,34,35,36}

22

Example

Pick (19,20)

Start

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

End

S
{1,2,7,8,9,13,19
14,20,26,27}
{3}
{4}
{5}
{6}
{10}
{11,17}
{12}
{15,16,21}
.
.
{22,23,24,29,39,32
33,34,35,36}

23

Example at the End

Start

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

End

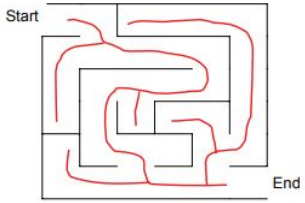
S
{1,2,3,4,5,6,7,… 36}

Remaining edges in E
Previously added to Maze

24

lab 7. Maze

A Hidden Tree



17

Number the Cells

We start with disjoint sets S ={ {1}, {2}, {3}, {4},… {36} }.
We have all possible edges between neighbors
E ={ (1,2), (1,7), (2,8), (2,3), … } 60 edges total.

Start

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

End

Idea: Union-find operations will be done on cells.

18

Maze Building with Disjoint Union/Find

Algorithm sketch:

1. Choose edge at random.
   → *Boundary edges are not in edge list, so left alone*
2. Erase it (and its wall) if the neighbors are in disjoint sets.
   → *Avoids cycles*
3. Take union of those sets.
4. Go to 1, iterate until there is only one set.
   → *Every cell reachable from every other cell.*

19

Pseudocode

- S = set of sets of connected cells
- E = set of edges
- Maze = set of maze edges initially empty

```
While there is more than one set in S
  Pick a random edge (x,y) and remove from E
  u = Find(x);
  v = Find(y);
  if u ≠ v then
    Union(u,v)
  else
    Add edge (x,y) to Maze
All remaining members of E together with Maze form
the maze
```

20

lab 7. Maze

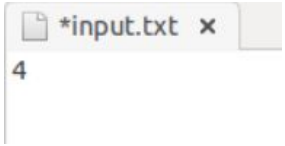
- **Random Number Generation**
  - Use srand() and rand() fuctions in <stdlib.h> and time() function in <time.h>

    exmaple

```
#include <stdlib.h>
#include <time.h>
srand((unsigned int)time(NULL)); // generate seed
...
int x = rand() // rand() function returns integer from 0 to 32767
...
int y = rand()%10 // y is from 0 to 9
```

lab 7. Maze

- You have to use file I/O like the previous assignment.
  - Input



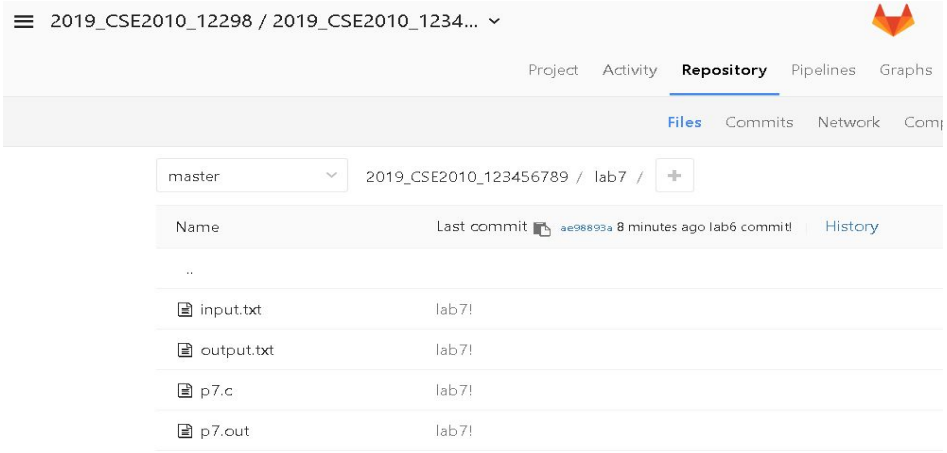
  - Output

```
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab7$ gcc p7.c -o p7.out
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab7$ ./p7.out
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab7$ cat output.txt
+-+-+-+-+
      | |
+ +-+-+ +
| | |   |
+ + +-+ +
|   |   |
+-+ + + +
|     |
+-+-+-+-+
```

The above result is only an example, and your results can always change.

lab 7. Maze

- Submission
  - Project directory name : lab7
  - Source file name : p7.c
  - Executable file name : p7.out
  - You should upload in the hconnect (Gitlab) server. **Please upload input, output file!**



DeadLine

Wednesday, 24 April, 23 : 59 pm