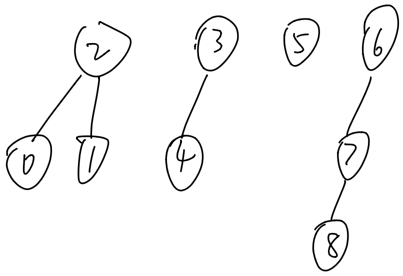
Assignment #8

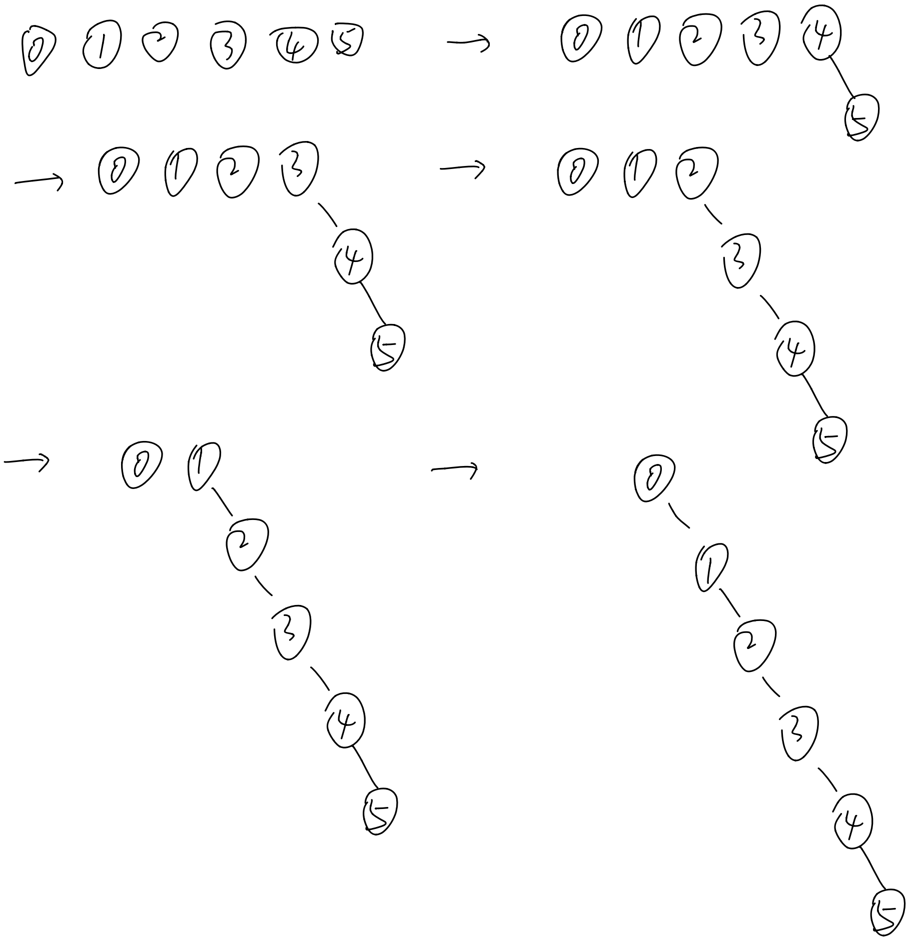
10 points
1) Given the array below representing disjoint sets, draw the associated trees.

2	2	-1	-1	3	-1	-1	6	7
0	1	2	3	4	5	6	7	8



15 points
2) Using a set of values from 0 to 5 as separate roots, perform the following unions by
making the root of the second tree be a child of the root of the first tree.
What do you notice about the resulting tree and what consequence would it have on a find(5)?

```
   union(4, 5)
   union(3, 4)
   union(2, 3)
   union(1, 2)
   union(0, 1)
```



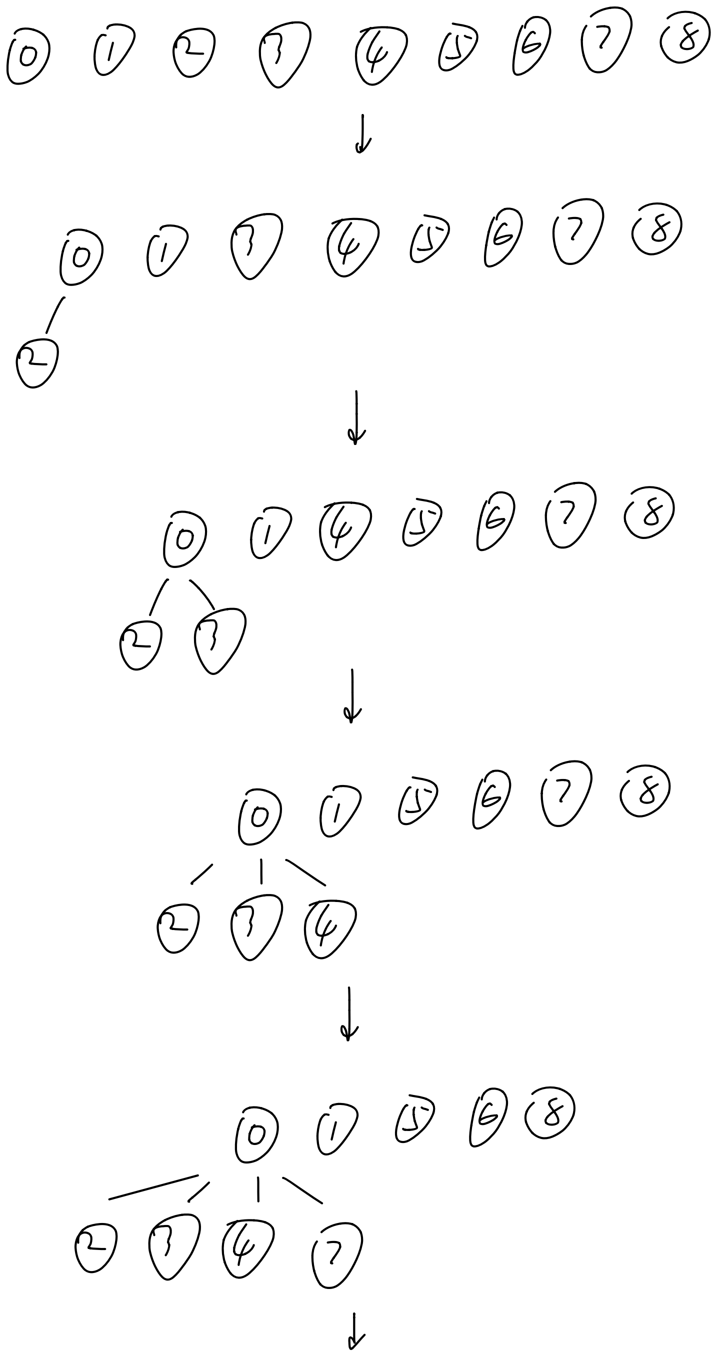
The resulting tree looks like a linked list and to find(5), it will traverse the whole tree, which takes linear time.

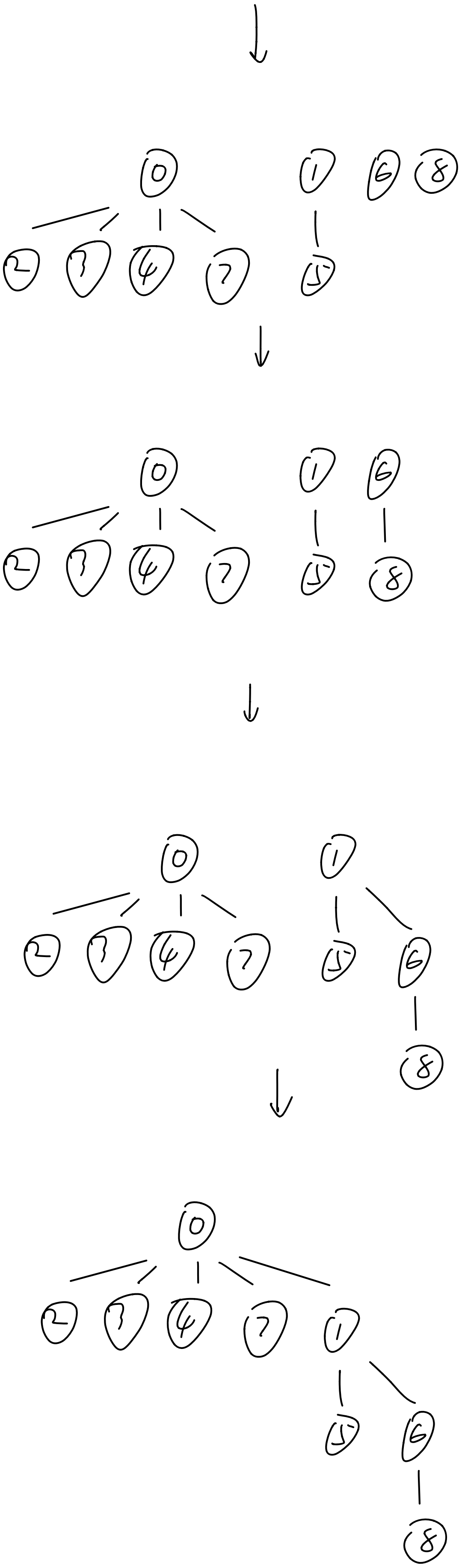
15 points

3) Using a set of values from 0 to 8 as separate roots, perform the following unions using union-by-size. Show the result of each union. When sizes are the same, make the second tree be a child of the first tree.

Notice the finds return roots, and a union will union two roots.

```
union(find(0),find(2))
union(find(0),find(3))
union(find(0),find(4))
union(find(0),find(7))
union(find(1),find(5))
union(find(6),find(8))
union(find(5),find(8))
union(find(7),find(8))
```




0
1
6
8
2
3
4
7
5
0
1
6
2
3
4
7
5
8
0
1
2
3
4
7
5
6
8
0
2
3
4
7
1
5
6
8

10 points

4) Illustrate the array for the final forest of the previous problem (note that

roots are not simply -1 when using union-by-size).

-1	-1	-1	-1	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-2	-1	0	-1	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-3	-1	0	0	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-4	-1	0	0	0	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-5	-1	0	0	0	-1	-1	0	-1
0	1	2	3	4	5	6	7	8

↓

-5	-2	0	0	0	1	-1	0	-1
0	1	2	3	4	5	6	7	8

↓

-5	-2	0	0	0	1	-2	0	6
0	1	2	3	4	5	6	7	8

↓

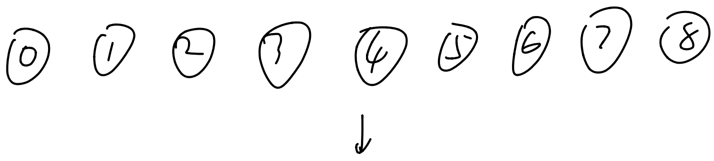
-5	-4	0	0	0	1	1	0	6
0	1	2	3	4	5	6	7	8

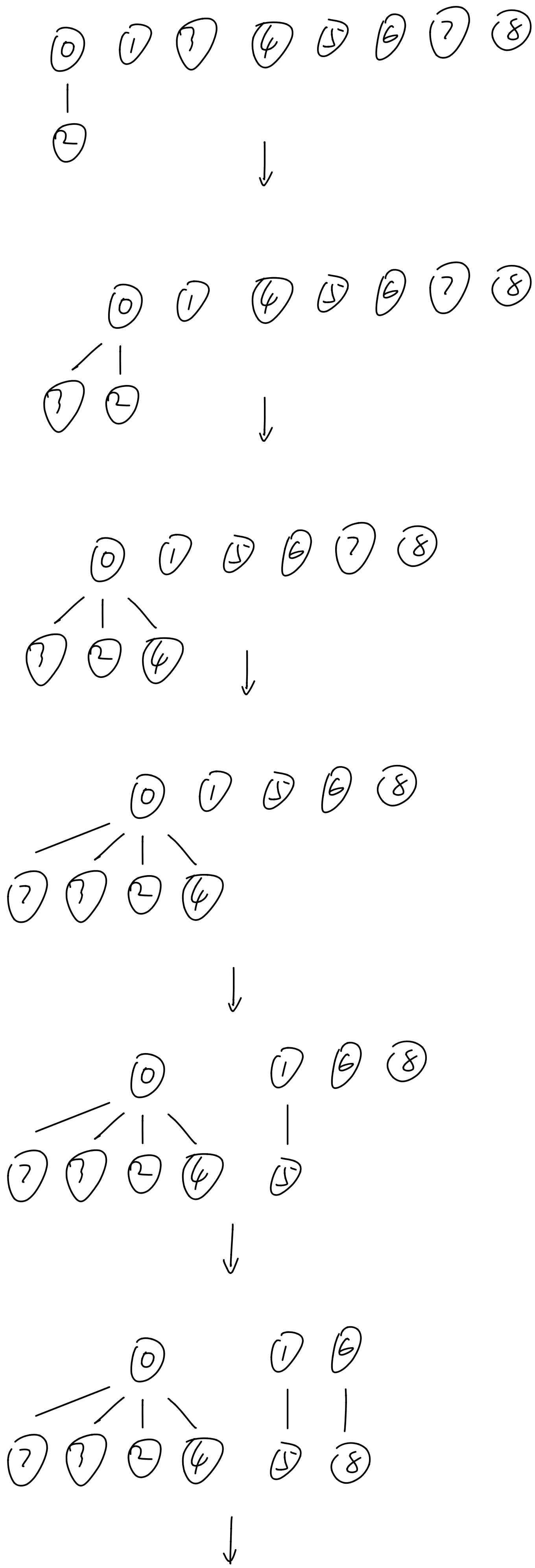
↓

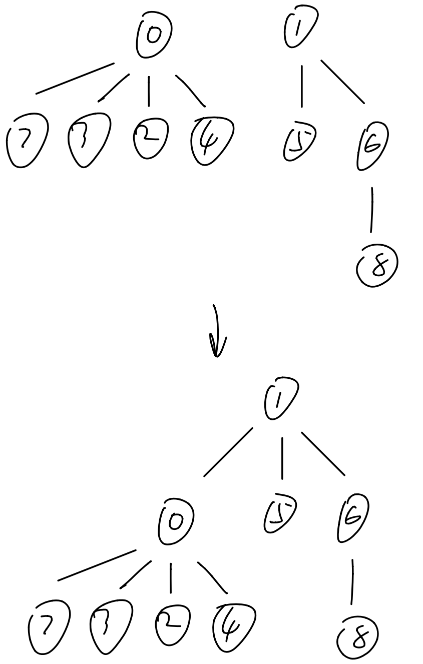
-9	0	0	0	0	1	1	0	6
0	1	2	3	4	5	6	7	8

15 points

5) Same as #3, but using union-by-height. When heights are the same, make the second tree be a child of the first tree.




0 1 3 4 5 6 7 8
2
0 1 4 5 6 7 8
3 2
0 1 5 6 7 8
3 2 4
0 1 5 6 8
7 3 2 4
0 1 6 8
7 3 2 4 5
0 1 6
7 3 2 4 5 8



10 points
6) Illustrate the array for the final forest of the previous problem (note that roots are not simply -1 when using union-by-height).

-1	-1	-1	-1	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-2	-1	0	-1	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-2	-1	0	0	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-2	-1	0	0	0	-1	-1	-1	-1
0	1	2	3	4	5	6	7	8

↓

-2	-1	0	0	0	-1	-1	0	-1
0	1	2	3	4	5	6	7	8

-2	-2	0	0	0	1	-1	0	-1
0	1	2	3	4	5	6	7	8

↓

-2	-2	0	0	0	1	-2	0	6
0	1	2	3	4	5	6	7	8

↓

-2	-3	0	0	0	1	1	0	6
0	1	2	3	4	5	6	7	8

↓

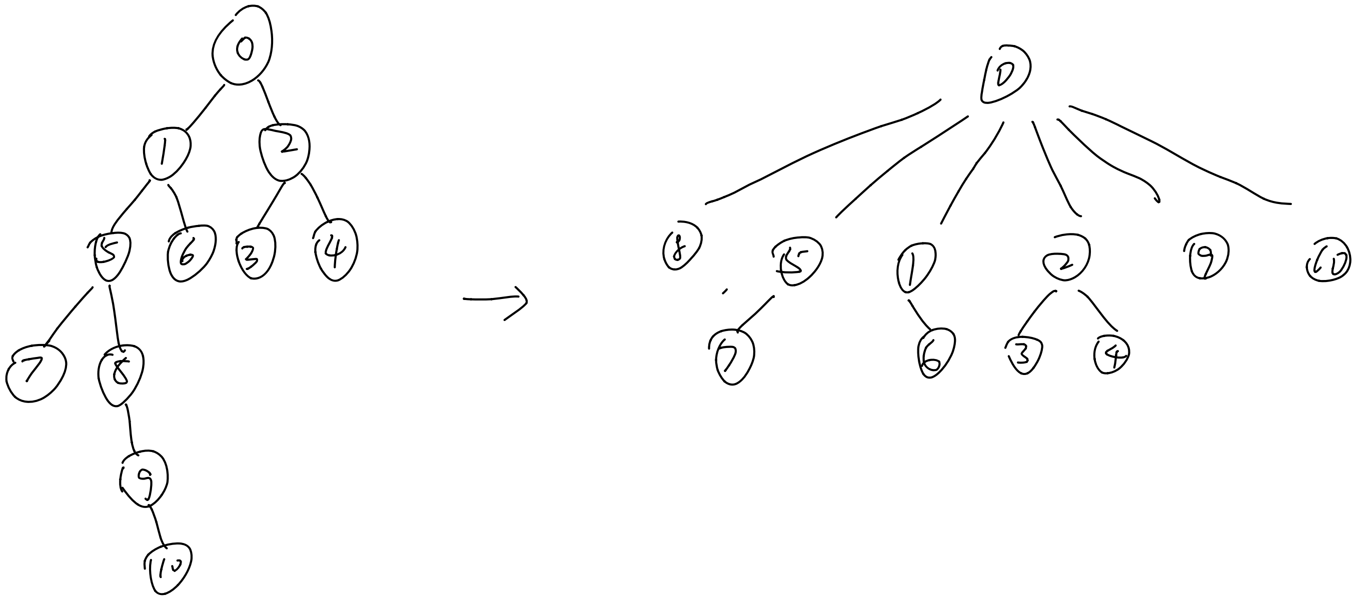
1	-3	0	0	0	1	1	0	6
0	1	2	3	4	5	6	7	8

15 points
7) Given the disjoint set array shown, what would the array look like after a find(10) if path compression is used?

-1	0	0	2	2	1	1	5	5	8	9
0	1	2	3	4	5	6	7	8	9	10

-1	0	0	2	2	0	1	5	0	0	0
0	1	2	3	4	5	6	7	8	9	10

10 points
8) Illustrate the trees for the final forest of the previous problem.


0
1
2
5
6
3
4
7
8
9
10
0
8
5
1
2
9
10
7
6
3
4