

Homework 1

1. Question 1

What are the best-case and worst-case tree heights for weighted quick-union and weighted quick-union with path compression? Give your answers in terms of order of growth.

Weighted QU

Worst case: height is $\log N$

Best case: 1

2. Question 2

Textbook: 1.5.1, 1.5.2, 1.5.3

1.5.1 Show the contents of the `id[]` array and the number of times the array is accessed for each input pair when you use quick-find for the sequence 9-0 3-4 5-8 7-2 2-1 5-7 0-3 4-2.

Initial array

Index: 0 1 2 3 4 5 6 7 8 9

Value: 0 1 2 3 4 5 6 7 8 9

9-0

2 array accesses for `connected()` and 21 for `union()`

Index: **0** 1 2 3 4 5 6 7 8 **9**

Value: **0** 1 2 3 4 5 6 7 8 **0**

3-4

2 array accesses for `connected()` and 21 for `union()`

Index: 0 1 2 **3** **4** 5 6 7 8 9

Value: 0 1 2 **4** **4** 5 6 7 8 0

5-8

2 array accesses for `connected()` and 21 for `union()`

Index: 0 1 2 3 4 **5** 6 7 **8** 9

Value: 0 1 2 4 4 **8** 6 7 **8** 0

7-2

2 array accesses for `connected()` and 21 for `union()`

Index: 0 1 **2** 3 4 5 6 **7** 8 9

Value: 0 1 **2** 4 4 8 6 **2** 8 0

2-1

2 array accesses for connected() and 21 for union()

Index: 0 **1** **2** 3 4 5 6 **7** 8 9

Value: 0 **1** **1** 4 4 8 6 **1** 8 0

5-7

2 array accesses for connected() and 21 for union()

Index: 0 **1** **2** 3 4 **5** 6 **7** 8 9

Value: 0 **1** **1** 4 4 **1** 6 **1** 8 0

0-3

2 array accesses for connected() and 21 for union()

Index: **0** 1 2 **3** 4 5 6 7 8 **9**

Value: **4** 1 1 **4** 4 1 6 1 8 **4**

4-2

2 array accesses for connected() and 21 for union()

Index: **0** 1 **2** **3** **4** 5 6 7 8 **9**

Value: **1** 1 **1** **1** **1** 1 6 1 8 **1**

1.5.2 Same as 1.5.1 but use quick-union. In addition, draw the forest of trees represented by the id[] array after each input pair is processed

(Need to add number of array accesses)

Initial array

Index: 0 1 2 3 4 5 6 7 8 9

Value: 0 1 2 3 4 5 6 7 8 9

Tree

0 1 2 3 4 5 6 7 8 9

9-0

2 array accesses for connected() and 3 for union()

Index: **0** 1 2 3 4 5 6 7 8 **9**

Value: **0** 1 2 3 4 5 6 7 8 **0**

Tree

0 1 2 3 4 5 6 7 8

|

9

3-4

2 array accesses for connected() and 3 for union()

Index: 0 1 2 **3** **4** 5 6 7 8 9

Value: 0 1 2 **4** **4** 5 6 7 8 0

Tree

0 1 2 4 5 6 7 8

|

9

|

3

5-8

2 array accesses for connected() and 3 for union()

Index: 0 1 2 3 4 **5** 6 7 **8** 9

Value: 0 1 2 4 4 8 6 7 8 0

Tree

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

7-2

2 array accesses for connected() and 3 for union()

Index: 0 1 **2** 3 4 5 6 **7** 8 9

Value: 0 1 **2** 4 4 8 6 **2** 8 0

Tree

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

2-1

2 array accesses for connected() and 3 for union()

Index: 0 **1** **2** 3 4 5 6 7 8 9

Value: 0 **1** **1** 4 4 8 6 2 8 0

Tree

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

5-7

8 array accesses for connected() and 9 for union()

Index: 0 1 **2** 3 4 **5** 6 **7** **8** 9

Value: 0 1 **1** 4 4 **8** 6 **2** **1** 0

Tree

0 1 4 6
| / \ |
9 2 8 3
| |
7 5

0-3

4 array accesses for connected() and 5 for union()

Index: **0** 1 2 **3** **4** 5 6 7 8 9

Value: **4** 1 1 **4** **4** 8 6 2 1 0

Tree

1 4 6
/ \ / \
2 8 3 0
| | |
7 5 9

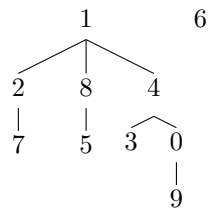
4-2

4 array accesses for connected() and 5 for union()

Index: 0 **1** **2** 3 **4** 5 6 7 8 9

Value: 4 **1** **1** 4 **1** 8 6 2 1 0

Tree



1.5.3 Same as 1.5.1 but use weighted quick-union

Initial array

Index: 0 1 2 3 4 5 6 7 8 9

Value: 0 1 2 3 4 5 6 7 8 9

Tree

0 1 2 3 4 5 6 7 8 9

9-0

Index: **0** 1 2 3 4 5 6 7 8 **9**

Value: **0** 1 2 3 4 5 6 7 8 **0**

Tree

0 1 2 3 4 5 6 7 8

|

9

3-4

Index: 0 1 2 **3** **4** 5 6 7 8 9

Value: 0 1 2 **4** **4** 5 6 7 8 0

Tree

0 1 2 4 5 6 7 8

|

9

|

3

5-8

Index: 0 1 2 3 4 **5** 6 7 **8** 9

Value: 0 1 2 4 4 **8** 6 7 **8** 0

Tree

0 1 2 4 6 7 8

|

9

|

3

|

5

7-2

Index: 0 1 **2** 3 4 5 6 **7** 8 9

Value: 0 1 **2** 4 4 8 6 **2** 8 0

Tree

0 1 2 4 6 8

|

9

|

7

|

3

|

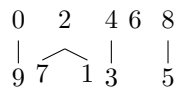
5

2-1

Index: 0 **1** **2** 3 4 5 6 7 8 9

Value: 0 **2** **2** 4 4 8 6 2 8 0

Tree

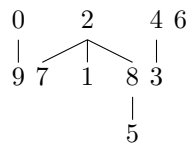


5-7

Index: 0 1 **2** 3 4 **5** 6 **7** **8** 9

Value: 0 2 **2** 4 4 **8** 6 **2** **2** 0

Tree

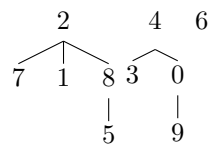


0-3

Index: **0** 1 2 **3** **4** 5 6 7 8 9

Value: **4** 2 2 **4** **4** 8 6 2 2 0

Tree



4-2

Index: 0 **1** **2** 3 **4** 5 6 7 8 9

Value: 4 **2** **2** 4 **1** 8 6 2 2 0

Tree

