

Program Structures and Algorithms

Spring 2023(SEC 1)

NAME: Harshil Shah

NUID: 002780887

Assignment: 4

Task: Your task is

Step 1:

(a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you have to do is to fill in the sections marked with // TO BE IMPLEMENTED ... // ...END IMPLEMENTATION.

(b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).

Step 2:

Using your implementation of UF_HWQUPC, develop a UF ("union-find") client that takes an integer value n from the command line to determine the number of "sites." Then generates random pairs of integers between 0 and $n-1$, calling `connected()` to determine if they are connected and `union()` if not. Loop until all sites are connected then print the number of connections generated. Package your program as a static method `count()` that takes n as the argument and returns the number of connections; and a `main()` that takes n from the command line, calls `count()` and prints the returned value. If you prefer, you can create a main program that doesn't require any input and runs the experiment for a fixed set of n values. Show evidence of your run(s).

Step 3:

Determine the relationship between the number of objects (n) and the number of pairs (m) generated to accomplish this (i.e. to reduce the number of components from n to 1). Justify your conclusion in terms of your observations and what you think might be going on.

Relationship Conclusion:

1. The number of pairs (m) generated to reduce the number of components from n to 1 is proportional to $n \log n$, because the union operation needs to be performed $\log n$ times for each of the n objects to reduce the number of components to 1.
2. However, the actual number of pairs generated may be less than $n \log n$ in practice, because the height-weighted Quick Union Find with path compression algorithm uses path compression to reduce the height of the trees. This results in a smaller number of nodes in the trees, which in turn reduces the number of union operations required to reduce the number of components to 1.
3. Therefore, the relationship between the number of objects (n) and the number of pairs (m) generated to accomplish the reduction of components from n to 1 is proportional to $n \log n$, but the actual number of pairs generated may be less due to the use of path compression.

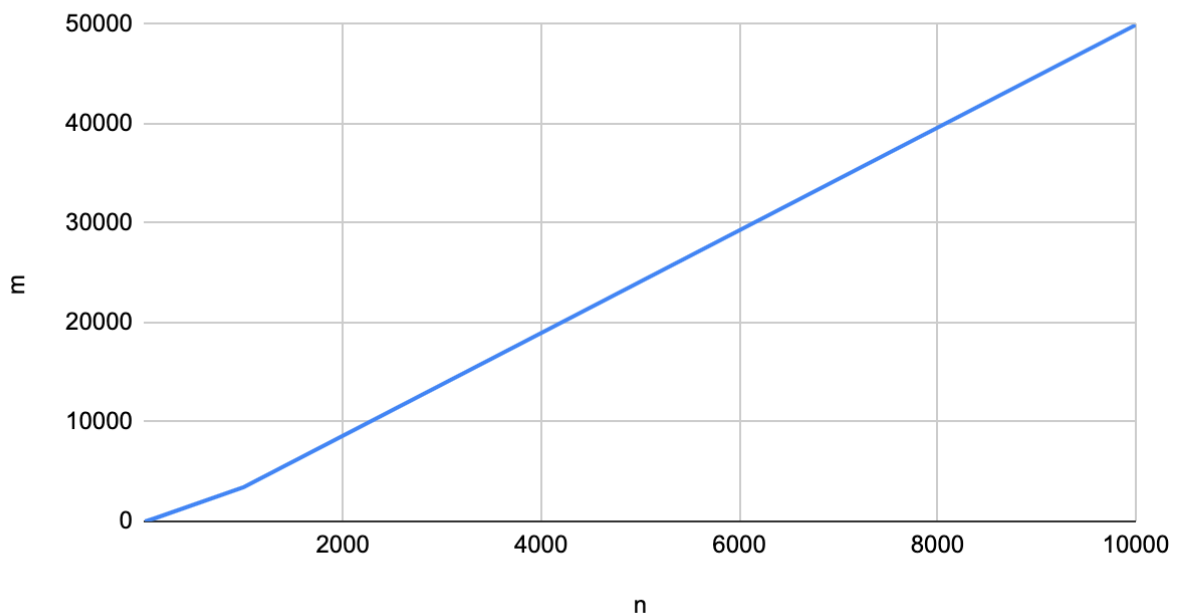
Evidence to support that conclusion:

n	m
1	0
10	11
100	290
1000	3430
10000	49935

Observation: These observations show that the height-weighted Quick Union Find with path compression algorithm is efficient for reducing the number of components from n to 1, but the time complexity increases as the number of objects grows. The use of path compression helps to reduce the time complexity, but the overall time complexity still grows logarithmically with the number of objects.

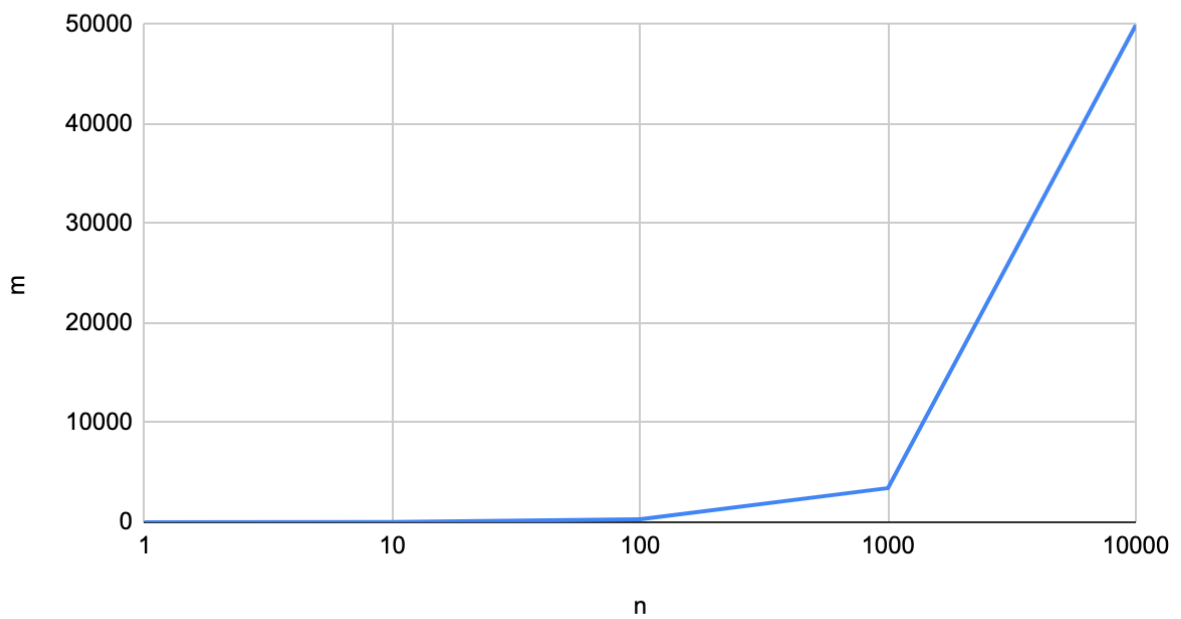
Graphical Representation:

m vs n



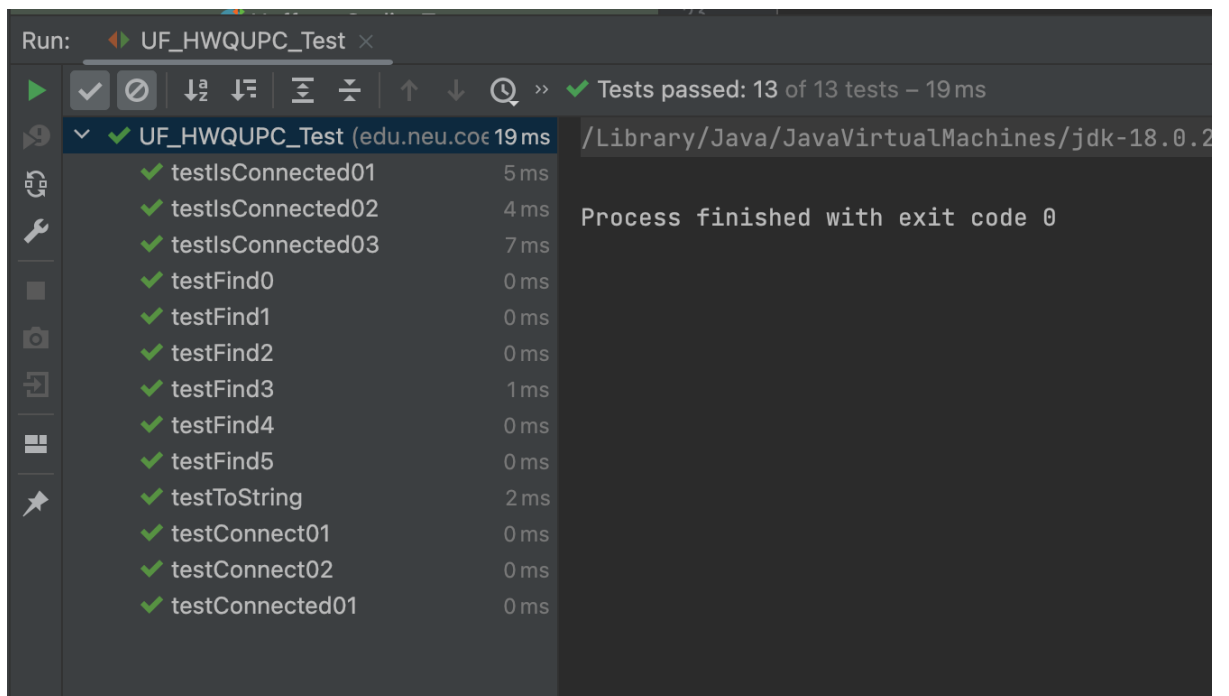
With Logarithmic Scale

m vs n



Unit Test Screenshots:

1. Unit test cases for UF_HWQUPC



> dynamicProgramming

> functions

> graphs

> greedy

> lab_1

> life

> pq

> randomwalk

> reduction

> sort

> symbolTable

> threesum

> union_find

UF_HWQUPC_Test

WQUPCTest

> util

BinarySearchTest

ComparableTupleTest

11

no usages xiaohuanlin

12 public class UF_HWQUPC_Test {

13

14 xiaohuanlin

15 @Test

16 public void testToString() {

17 Connections h = new UF_HWQUPC(n: 2);

18 assertEquals(expected: "UF_HWQUPC:\n" +

19 " count: 2\n" +

20 " path compression? true\n" +

21 " parents: [0, 1]\n" +

22 " heights: [1, 1]", h.toString());

23 }

24 /**

25 *

Run: UF_HWQUPC_Test.testToString x

Tests passed: 1 of 1 test – 16 ms

UF_HWQUPC_Test (edu.neu.coe 16 ms)

testToString 16 ms

Process finished with exit code 0

> functions

> graphs

> greedy

> lab_1

> life

> pq

> randomwalk

> reduction

> sort

> symbolTable

> threesum

> union_find

UF_HWQUPC_Test

WQUPCTest

> util

BinarySearchTest

ComparableTupleTest

24 /**

25 *

26 */

27 no usages xiaohuanlin

28 @Test

29 public void testIsConnected01() {

30 Connections h = new UF_HWQUPC(n: 2);

31 assertFalse(h.isConnected(p: 0, q: 1));

32 }

33 /**

34 *

35 */

36 no usages xiaohuanlin

@Test(expected = IllegalArgumentException.class)

Run: UF_HWQUPC_Test.testIsConnected01 x

Tests passed: 1 of 1 test – 3 ms

UF_HWQUPC_Test (edu.neu.coe 3 ms)

testIsConnected01 3 ms

Process finished with exit code 0

life

pq

randomwalk

reduction

sort

symbolTable

threesum

union_find

UF_HWQUPC_Test

WQUPCTest

util

BinarySearchTest

ComparableTupleTest

35

36

37

38

39

40

41

42

43

44

no usages xiaohuanlin

@Test(expected = IllegalArgumentException.class)

public void testIsConnected02() {

Connections h = new UF_HWQUPC(n: 1);

assertTrue(h.isConnected(p: 0, q: 1));

}

/**

*

*/

no usages xiaohuanlin

run: UF_HWQUPC_Test.testIsConnected02 x

Tests passed: 1 of 1 test - 5 ms

UF_HWQUPC_Test (edu.neu.coe. 5 ms)

testIsConnected02 5 ms

Process finished with exit code 0

functions

graphs

greedy

lab_1

life

pq

randomwalk

reduction

sort

symbolTable

threesum

union_find

UF_HWQUPC_Test

WQUPCTest

util

BinarySearchTest

ComparableTupleTest

45

46

47

48

49

50

51

52

53

54

55

56

57

58

no usages xiaohuanlin

@Test

public void testIsConnected03() {

Connections h = new UF_HWQUPC(n: 2);

final PrivateMethodTester tester = new PrivateMethodTester(h);

assertNull(tester.invokePrivate(name: "updateParent", ...parameters: 0, 1));

assertTrue(h.isConnected(p: 0, q: 1));

}

/**

*

*/

no usages xiaohuanlin

@Test

public void testConnect01() {

Connections h = new UF_HWQUPC(n: 2);

run: UF_HWQUPC_Test.testIsConnected03 x

Tests passed: 1 of 1 test - 14 ms

UF_HWQUPC_Test (edu.neu.coe. 14 ms)

testIsConnected03 14 ms

Process finished with exit code 0

graphs

greedy

lab_1

life

pq

randomwalk

reduction

sort

symbolTable

threesum

union_find

UF_HWQUPC_Test

WQUPCTest

util

BinarySearchTest

ComparableTupleTest

56

57

58

59

60

61

62

63

64

65

66

67

no usages xiaohuanlin

@Test

public void testConnect01() {

Connections h = new UF_HWQUPC(n: 2);

h.connect(p: 0, q: 1);

}

/**

*

*/

no usages xiaohuanlin

@Test

public void testConnect02() {

Connections h = new UF_HWQUPC(n: 2);

run: UF_HWQUPC_Test.testConnect01 x

Tests passed: 1 of 1 test - 2 ms

UF_HWQUPC_Test (edu.neu.coe. 2 ms)

testConnect01 2 ms

Process finished with exit code 0

functionsgraphsgreedylab_1lifepqrandomwalkreductionsortsymbolTablethreesumunion_findUF_HWQUPC_TestWQUPCTestutilBinarySearchTestComparableTupleTest

UF_HWQUPC_Test.testConnect02

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

```
@Test
public void testConnect02() {
    Connections h = new UF_HWQUPC( n: 2);
    h.connect( p: 0, q: 1);
    h.connect( p: 0, q: 1);
    assertTrue(h.isConnected( p: 0, q: 1));
}

/**
 *
 */
no usages xiaohuanlin

@Test
public void testFind0() {
    UF h = new UF_HWQUPC( n: 1);
    assertEquals( expected: 0, h.find( p: 0));
```

Tests passed: 1 of 1 test - 2 ms

UF_HWQUPC_Test (edu.neu.coe. 2 ms)

testConnect02 2 ms

Process finished with exit code 0

graphsgreedylab_1lifepqrandomwalkreductionsortsymbolTablethreesumunion_findUF_HWQUPC_TestWQUPCTestutilBinarySearchTestComparableTupleTest

UF_HWQUPC_Test.testFind0

76

77

78

79

80

81

82

83

84

85

86

87

88

89

```
@Test
public void testFind0() {
    UF h = new UF_HWQUPC( n: 1);
    assertEquals( expected: 0, h.find( p: 0));
}

/**
 *
 */
no usages xiaohuanlin

@Test
public void testFind1() {
    UF h = new UF_HWQUPC( n: 2);
    h.connect( p: 0, q: 1);
    assertEquals( expected: 0, h.find( p: 0));
```

Tests passed: 1 of 1 test - 2 ms

UF_HWQUPC_Test (edu.neu.coe. 2 ms)

UF_HWQUPC_Test.testFind0 2 ms

Process finished with exit code 0

functionsgraphsgreedylab_1lifepqrandomwalkreductionsortsymbolTablethreesumunion_findUF_HWQUPC_TestWQUPCTestutilBinarySearchTestComparableTupleTest

UF_HWQUPC_Test.testFind1

```
82  /**
83  *
84  */
85  no usages xiaohuanlin
86  @Test
87  public void testFind1() {
88      UF h = new UF_HWQUPC( n: 2);
89      h.connect( p: 0, q: 1);
90      assertEquals( expected: 0, h.find( p: 0));
91      assertEquals( expected: 0, h.find( p: 1));
92  }
93  /**
94  *
95  */
96  no usages xiaohuanlin
```

Run: UF_HWQUPC_Test.testFind1

Tests passed: 1 of 1 test – 3 ms

UF_HWQUPC_Test (edu.neu.coe. 3 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testFind1 3 ms

Process finished with exit code 0

dynamicProgrammingfunctionsgraphsgreedylab_1lifepqrandomwalkreductionsortsymbolTablethreesumunion_findUF_HWQUPC_TestWQUPCTestutilBinarySearchTestComparableTupleTest

UF_HWQUPC_Test.testFind2

```
96  no usages xiaohuanlin
97  @Test
98  public void testFind2() {
99      UF h = new UF_HWQUPC( n: 3, pathCompression: false);
100      h.connect( p: 0, q: 1);
101      assertEquals( expected: 0, h.find( p: 0));
102      assertEquals( expected: 0, h.find( p: 1));
103      h.connect( p: 2, q: 1);
104      assertEquals( expected: 0, h.find( p: 0));
105      assertEquals( expected: 0, h.find( p: 1));
106      assertEquals( expected: 0, h.find( p: 2));
107  }
108  /**
109  *
110  */
111  no usages xiaohuanlin
```

Run: UF_HWQUPC_Test.testFind2

Tests passed: 1 of 1 test – 2 ms

UF_HWQUPC_Test (edu.neu.coe. 2 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java

testFind2 2 ms

Process finished with exit code 0

```
edu.neu.coe.info6205
├── bqs
├── dynamicProgramming
├── functions
├── graphs
├── greedy
├── lab_1
├── life
├── pq
├── randomwalk
├── reduction
├── sort
├── symbolTable
├── threesum
└── union_find
    ├── UF_HWQUPC_Test
    ├── WQUPCTest
    └── util
        ├── BinarySearchTest
        ├── ComparableTupleTest
        ├── HuffmanCodingTest
        ├── MyDateTest
        ├── TailCallTest
        ├── TicketTest
        └── TupleTest
```

```
112 public void testFind3() {
113     UF h = new UF_HWQUPC( n: 6, pathCompression: false);
114     h.connect( p: 0, q: 1);
115     h.connect( p: 0, q: 2);
116     h.connect( p: 3, q: 4);
117     h.connect( p: 3, q: 5);
118     assertEquals( expected: 0, h.find( p: 0));
119     assertEquals( expected: 0, h.find( p: 1));
120     assertEquals( expected: 0, h.find( p: 2));
121     assertEquals( expected: 3, h.find( p: 3));
122     assertEquals( expected: 3, h.find( p: 4));
123     assertEquals( expected: 3, h.find( p: 5));
124     h.connect( p: 0, q: 3);
125     assertEquals( expected: 0, h.find( p: 0));
126     assertEquals( expected: 0, h.find( p: 1));
127     assertEquals( expected: 0, h.find( p: 2));
128     assertEquals( expected: 0, h.find( p: 3));
129     assertEquals( expected: 0, h.find( p: 4));
130     assertEquals( expected: 0, h.find( p: 5));
131     final PrivateMethodTester tester = new PrivateMethodTester(h);
132     assertEquals( expected: 3, tester.invokePrivate( name: "getParent", ...parameters: 4));
133     assertEquals( expected: 3, tester.invokePrivate( name: "getParent", ...parameters: 5));
134 }
```

Run: UF_HWQUPC_Test.testFind3 ×

Tests passed: 1 of 1 test – 12 ms

UF_HWQUPC_Test (edu.neu.coe.info6205) 12 ms

testFind3 12 ms

Process finished with exit code 0

```
edu.neu.coe.info6205
├── bqs
├── dynamicProgramming
├── functions
├── graphs
├── greedy
├── lab_1
├── life
├── pq
├── randomwalk
├── reduction
├── sort
├── symbolTable
├── threesum
└── union_find
    ├── UF_HWQUPC_Test
    ├── WQUPCTest
    └── util
        ├── BinarySearchTest
        ├── ComparableTupleTest
        ├── HuffmanCodingTest
        ├── MyDateTest
        ├── TailCallTest
        ├── TicketTest
        └── TupleTest
```

```
140 public void testFind4() {
141     UF h = new UF_HWQUPC( n: 6);
142     h.connect( p: 0, q: 1);
143     h.connect( p: 0, q: 2);
144     h.connect( p: 3, q: 4);
145     h.connect( p: 3, q: 5);
146     assertEquals( expected: 0, h.find( p: 0));
147     assertEquals( expected: 0, h.find( p: 1));
148     assertEquals( expected: 0, h.find( p: 2));
149     assertEquals( expected: 3, h.find( p: 3));
150     assertEquals( expected: 3, h.find( p: 4));
151     assertEquals( expected: 3, h.find( p: 5));
152     h.connect( p: 0, q: 3);
153     assertEquals( expected: 0, h.find( p: 0));
154     assertEquals( expected: 0, h.find( p: 1));
155     assertEquals( expected: 0, h.find( p: 2));
156     assertEquals( expected: 0, h.find( p: 3));
157     assertEquals( expected: 0, h.find( p: 4));
158     assertEquals( expected: 0, h.find( p: 5));
159     final PrivateMethodTester tester = new PrivateMethodTester(h);
160     assertEquals( expected: 0, tester.invokePrivate( name: "getParent", ...parameters: 4));
161     assertEquals( expected: 0, tester.invokePrivate( name: "getParent", ...parameters: 5));
162 }
```

Run: UF_HWQUPC_Test.testFind4 ×

Tests passed: 1 of 1 test – 12 ms

UF_HWQUPC_Test (edu.neu.coe.info6205) 12 ms

testFind4 12 ms

Process finished with exit code 0


```
166  */
167  no usages  xiaohuanlin
168  @Test(expected = IllegalArgumentException.class)
169  public void testFind5() {
170      UF h = new UF_HWQUPC( n: 1);
171      h.find( p: 1);
172  }
173
174  /**
175   *
176   */
177  no usages  xiaohuanlin
178  @Test
179  public void testConnected01() {
180      Connections h = new UF_HWQUPC( n: 10);
181      h.show();
182      assertFalse(h.isConnected( p: 0, q: 1));
183  }
```

run: UF_HWQUPC_Test.testFind5 x

Tests passed: 1 of 1 test - 5 ms

UF_HWQUPC_Test (edu.neu.coe. 5 ms)

testFind5 5 ms

Process finished with exit code 0

```
174  /**
175   *
176   */
177  no usages  xiaohuanlin
178  @Test
179  public void testConnected01() {
180      Connections h = new UF_HWQUPC( n: 10);
181      h.show();
182      assertFalse(h.isConnected( p: 0, q: 1));
183  }
```

run: UF_HWQUPC_Test.testConnected01 x

Tests passed: 1 of 1 test - 3 ms

UF_HWQUPC_Test (edu.neu.coe. 3 ms)

testConnected01 3 ms

Process finished with exit code 0

UF_Client execution:

```
Run: HWQUPC_Solution x
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/
n = 1, Number of connections(m) : 0
n = 10, Number of connections(m) : 11
n = 100, Number of connections(m) : 290
n = 1000, Number of connections(m) : 3430
n = 10000, Number of connections(m) : 49935

Process finished with exit code 0
```