Difeng Li (NUID: 001050353)

INFO 6205

Program Structures & Algorithms

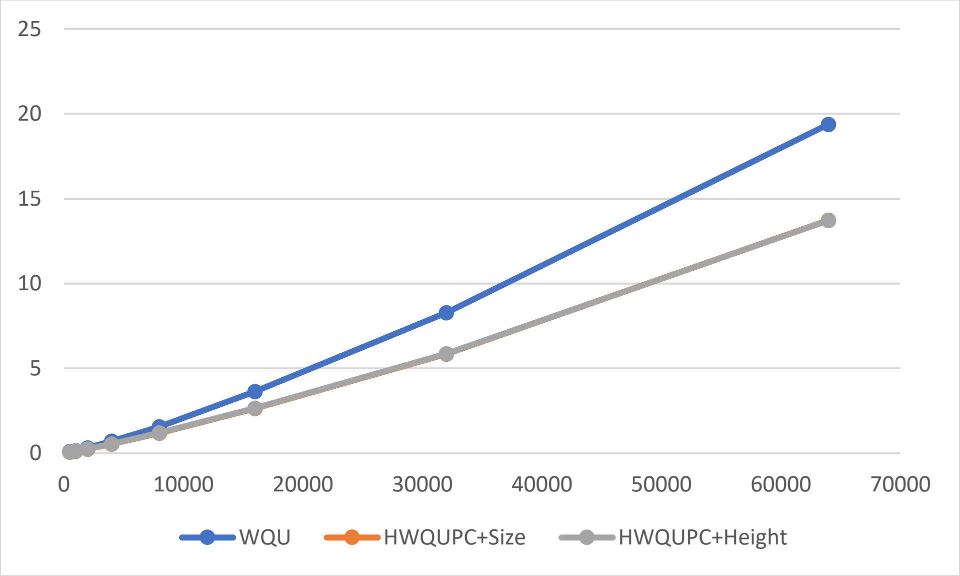
Fall 2020

Assignment No.4

- **Task:** **Quick Union with Path Compression with Benchmark**
- **Output:** The union find algorithm based on height after path compression is faster than the original algorithm without path compression.
- **Relationship conclusion**

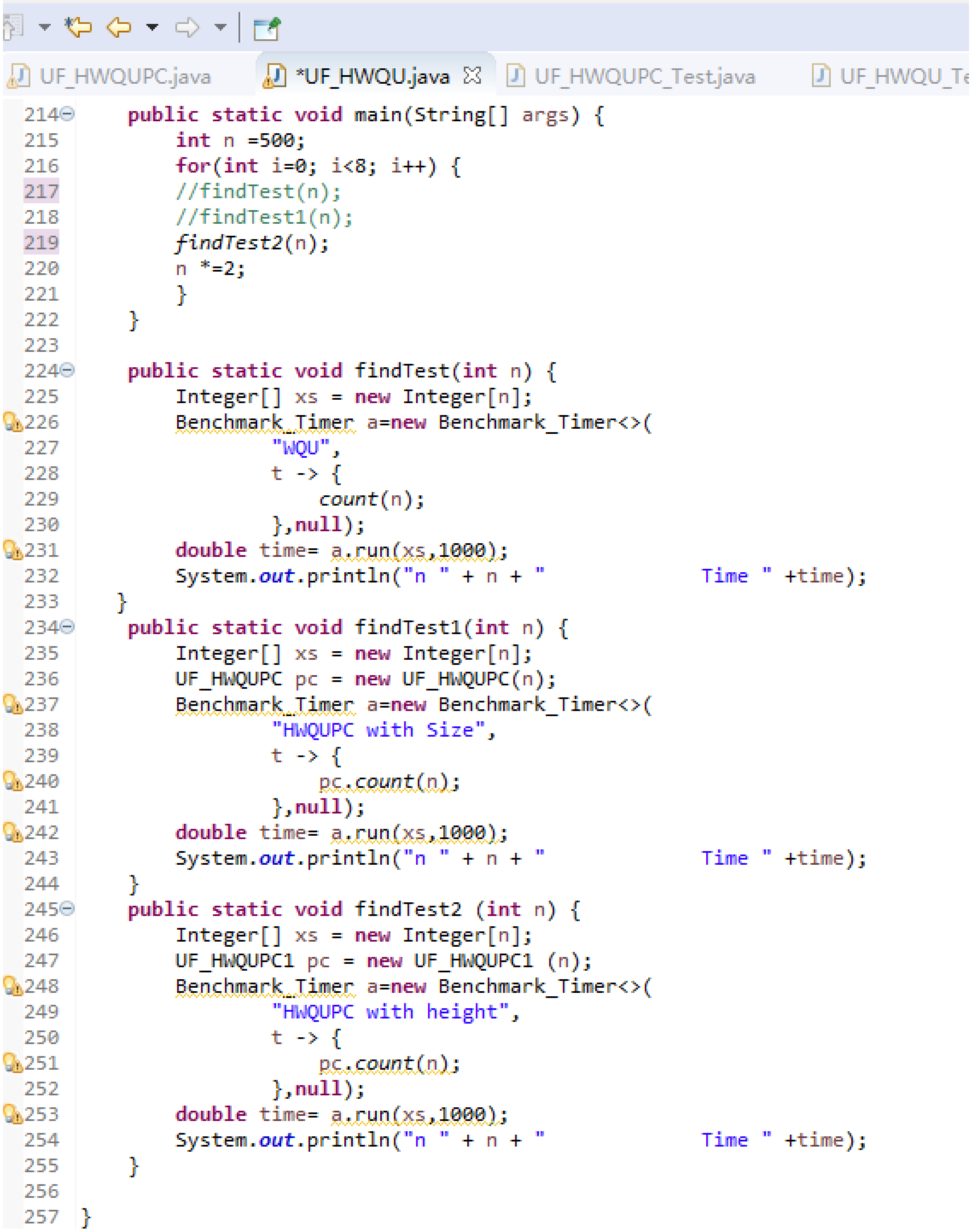
  The time complexity of find () function of height based quick union algorithm without path compression is O (lgN), and the time complexity of union () function is O (lgN).

  The time complexity of find () function of height based quick union algorithm with path compression is O (lgN), and the time complexity of union () function is O (n + Mlg * n).
- **Evidence to support relationship**



- ## Screenshot of Unit test passing

Test code



```java
    public static void main(String[] args) {
        int n =500;
        for(int i=0; i<8; i++) {
        //findTest(n);
        //findTest1(n);
        findTest2(n);
        n *=2;
        }
    }

    public static void findTest(int n) {
        Integer[] xs = new Integer[n];
        Benchmark_Timer a=new Benchmark_Timer<>(
                "WQU",
                t -> {
                    count(n);
                },null);
        double time= a.run(xs,1000);
        System.out.println("n " + n + "             Time " +time);
   }
    public static void findTest1(int n) {
        Integer[] xs = new Integer[n];
        UF_HWQUPC pc = new UF_HWQUPC(n);
        Benchmark_Timer a=new Benchmark_Timer<>(
                "HWQUPC with Size",
                t -> {
                    pc.count(n);
                },null);
        double time= a.run(xs,1000);
        System.out.println("n " + n + "             Time " +time);
    }
    public static void findTest2 (int n) {
        Integer[] xs = new Integer[n];
        UF_HWQUPC1 pc = new UF_HWQUPC1 (n);
        Benchmark_Timer a=new Benchmark_Timer<>(
                "HWQUPC with height",
                t -> {
                    pc.count(n);
                },null);
        double time= a.run(xs,1000);
        System.out.println("n " + n + "             Time " +time);
    }

}
```

UF_HWQUPC.java -> compass with size
UF_HWQUPC1.java -> compass with height
UF_HWQU -> without compass