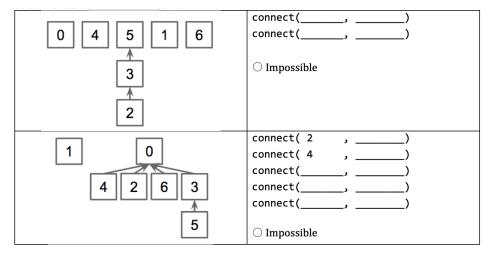# 1 Disjoint Sets

For each of the arrays below, write whether this could be the array representation of a weighted quick union with path compression and explain your reasoning.

```
    i:    0   1   2   3   4   5   6   7   8   9
    -----------------------------------------
A. a[i]:   1   2   3   0   1   1   1   4   4   5
B. a[i]:   9   0   0   0   0   0   9   9   9 -10
C. a[i]:   1   2   3   4   5   6   7   8   9 -10
D. a[i]: -10   0   0   0   0   1   1   1   6   2
E. a[i]: -10   0   0   0   0   1   1   1   6   8
F. a[i]:  -7   0   0   1   1   3   3  -3   7   7
```

2 Weighted Quick Union (Spring 2017, MT2)

Suppose we have a **weighted quick union** object. What calls to connect(a, b) produce the following trees? Assume that each **WQU** starts with all items disconnected. Fill in the "Impossible" option if the given tree is impossible. To tie break, the root of the left argument is placed below the root of the right argument.

0 4 5 1 6 3 2	connect(_____, _____) connect(_____, _____) ○ Impossible
1 0 4 2 6 3 5	connect(2 , _____) connect(4 , _____) connect(_____, _____) connect(_____, _____) connect(_____, _____) ○ Impossible

Suppose we add a new operation undo(a, b) that undoes an earlier Disjoint Sets connect operation. If connect(a, b) has never been called, then this method has no effect. For each of the implementations of Disjoint Sets, mark the corresponding box if it is impossible to add the undo operation without adding additional data structures (i.e. instance variables) to that implementation.

☐ Quick Union   ☐ Quick Find   ☐ Weighted Quick Union(WQU)   ☐ WQU with Path Compression

3   It Begins (Spring 2017, MT2)

For each code block below, fill in the blank(s) so that the function has the desired runtime. Do not use any commas. If the answer is impossible, just write "impossible" in the blank.

```
1  public static void f1(int N) {        //Desired Runtime: Θ(N)
2      for (int i = 1; i < N; _____) {System.out.println("hi");}
3  }
```

```
1  public static void f2(int N) {        //Desired Runtime: Θ(logN)
2      for (int i = 1; i < N; _____) {System.out.println("hi");}
3  }
```

```
1  public static void f3(int N) {        //Desired Runtime: Θ(1)
2      for (int i = 1; i < N; _____) {System.out.println("hi");}
3  }
```

4   Slightly Harder (Spring 2017, MT2)

Give the runtime of the following functions in Θ or O notation as requested. Your answer should be as simple as possible with no unnecessary leading constants or lower order terms. For f5, your bound should be as tight as possible (so don't just put $O(N^{NM!})$ or similar for the second answer).

```
1  Θ_____   public static void f4(int N) {
2                   if (N == 0) return;
3                   f4(N / 2);
4                   f4(N / 2);
5                   f4(N / 2);
6                   f4(N / 2);
7                   g(N); // runs in Θ(N²) time
8               }
```

```
1  O_____   public static void f5(int N, int M) {
2                   if (N < 10) return;
3                   for (int i = 0; i <= N % 10; i++) {
4                       f5(N / 10, M / 10);
5                       System.out.println(M);
6                   }
7               }
```