Shamar Webster

ID#1700087

Recursive

```
public int gcd(int x, int y) {

1 if (y == 0) {
1        return x;
  }
1 if (x>=y && x!=0) {
1         return gcd(y, x%y)   ;
  }
1        return 0;
}
```

1(1)+1(1)+1

1+1+1

3

O(1) **Constant**

```
public int ack(int x, int y) {

        1       if (x == 0) {
        1               y = y + 1;
        1               return y;
                }
        1       if (y == 0) {
        1               return ack(x - 1, 1);
                }
        1       if (x!=0 && y !=0) {
        1               return ack(x - 1, ack(x ,y- 1));
                }

        1       return 0;
        }
```

1(1+1)+1(1)+1(1)+1

1(2)+1+1+1

4

O(1) **Constant**

Iteratively

```
public int gcd(int x, int y) {

        logN    while(y != 0 && x >= y && x != 0) {
        1               y = x%y;
        1               return y;
                }
        1       return x;
        }
```

logN(1+1)+1

logN(2)+1

2logN+1

2logN+0

2logN

1logN

O(logN) **LOGARITHMIC**

```
public int ack(int x, int y) {

        N       while (x == 0) {
        1               y= y + 1;
        1               return y;
                }
        N       while (y == 0) {
        1               x = x - 1;
        1               return x, 1;
                }
        N       while (y != 0 && x >= y && x != 0){

        1               x = x -1;
        1               y = x, y-1;
                }
        1       return x;
        }
```

N(1+1)+N(1+1)+N(1+1)+1

N(2)+N(2)+N(2)+1

2N+2N+2N+1

6N+1

6N+0

6N

1N

O(N) **lINEAR**