

1. jede lebendige Zelle, die weniger als zwei lebendige Nachbarn hat, stirbt an Einsamkeit
2. jede lebendige Zelle mit mehr als drei lebendigen Nachbarn stirbt an Überbevölkerung
3. jede lebendige Zelle mit zwei oder drei Nachbarn fühlt sich wohl und lebt weiter
4. jede tote Zelle mit genau drei lebendigen Nachbarn wird wieder zum Leben erweckt

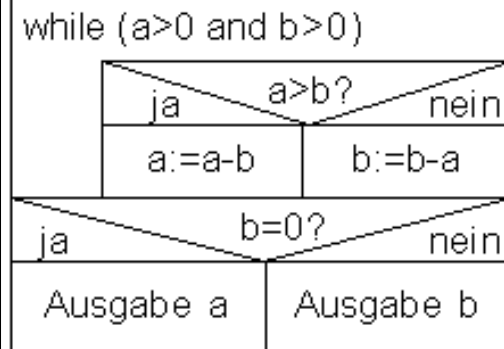
## GameOfLife

```
machWas(int[] meinArray) {
    for (int i = 0; i < 5; i++) {
        meinArray[i] = 0;
    }
}
```

$O(1)$

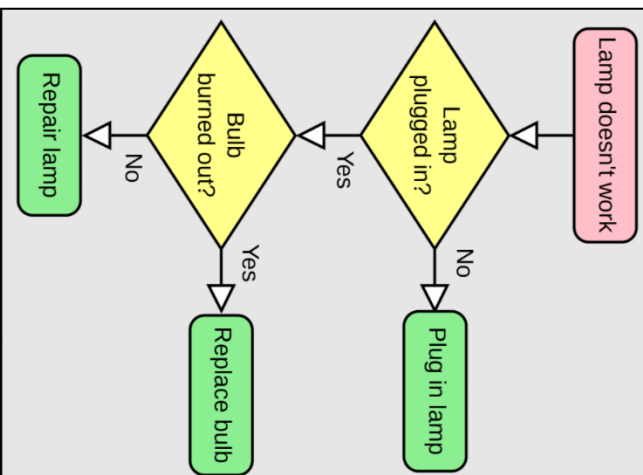
```
machWas(int[] meinArray) {
    for (int i = 0; i < meinArray.length; i++) {
        for (int j = 0; j < meinArray.length; j++) {
            meinArray[i] = 0;
        }
    }
}
```

$O(n^2)$



```
machWas(int[] meinArray) {
    for (int i = 0; i < meinArray.length; i++) {
        meinArray[i] = 0;
    }
}
```

$O(n)$



```
public static int[] sort(int[] arr) {
    boolean isSorted = false;
    while (!isSorted) {
        isSorted = true;
        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                int t = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = t;
                isSorted = false;
            }
        }
    }
    return arr;
}
```

BubbleSort

```
start = System.currentTimeMillis();
```

```
public static int[] sort(int[] arr) {
    int l = arr.length;
    for (int i = 1; i < l; i++) {
        int k = arr[i];
        for (int j = i - 1; j >= 0; j--) {
            if (arr[j] > k) {
                arr[j + 1] = arr[j];
                arr[j] = k;
            }
        }
    }
    return arr;
}
```

InsertionSort

```
void sort(int arr[])
{
    int n = arr.length;

    // One by one move boundary of unsorted subarray
    for (int i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        int min_idx = i;
        for (int j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        // Swap the found minimum element with the first
        // element
        int temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}
```

SelectionSort