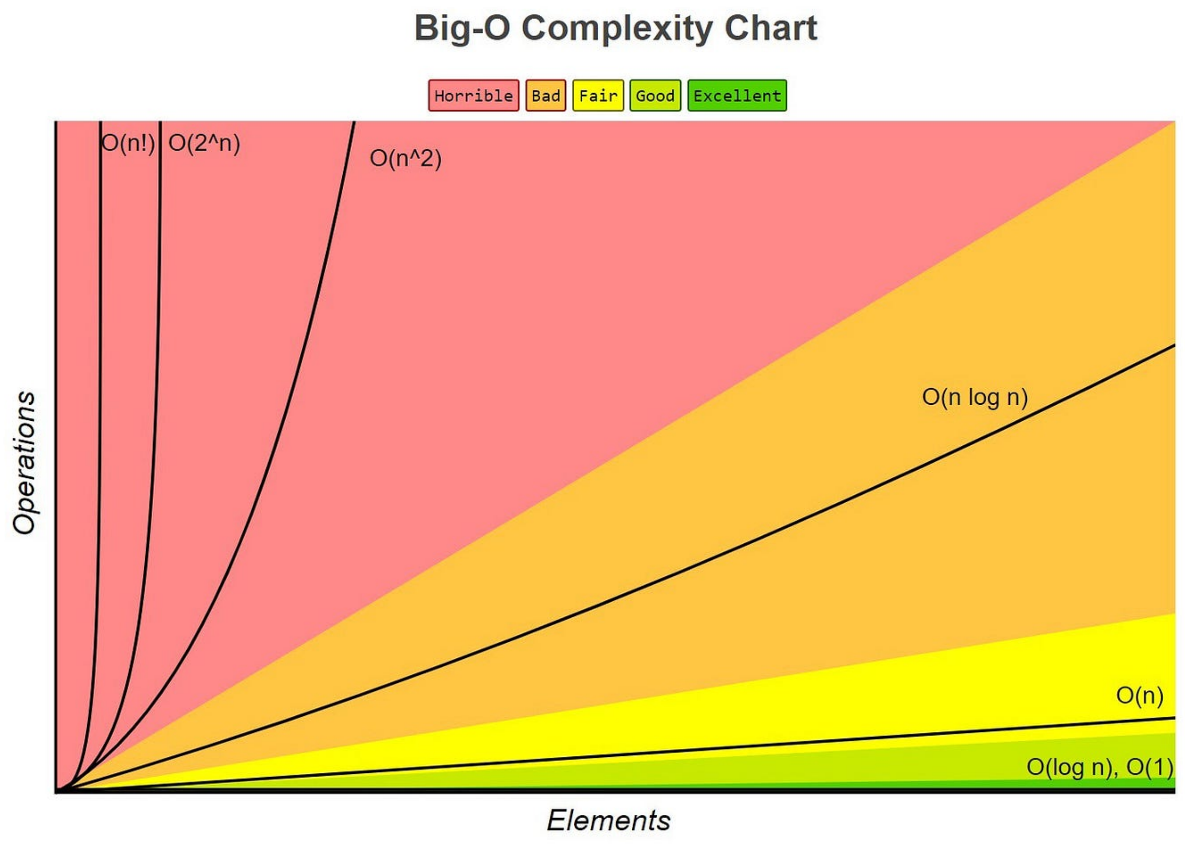
Recitation III: Asymptotic Complexity

Tags	
Date	@February 9, 2024
Files & media	DS recitation 3.pdf



```java
/** Returns true if there are no duplicate elements in the array. */
public static boolean unique1(int[ ] data) {
  int n = data.length;
  for (int j=0; j < n−1; j++)
    for (int k=j+1; k < n; k++)
      if (data[j] == data[k])
        return false;                          // found duplicate pair
  return true;                                 // if we reach this, elements are unique
}
```

Code Fragment 4.7: Algorithm unique1 for testing element uniqueness.

```java
/** Returns true if there are no duplicate elements in the array. */
public static boolean unique2(int[ ] data) {
  int n = data.length;
  int[ ] temp = Arrays.copyOf(data, n);        // make copy of data
  Arrays.sort(temp);                           // and sort the copy
  for (int j=0; j < n−1; j++)
    if (temp[j] == temp[j+1])                  // check neighboring entries
      return false;                            // found duplicate pair
  return true;                                 // if we reach this, elements are unique
}
```

Code Fragment 4.8: Algorithm unique2 for testing element uniqueness.

```java
/** Returns the sum of the prefix sums of given array. */
public static int example3(int[ ] arr) {
  int n = arr.length, total = 0;
  for (int j=0; j < n; j++)                              // loop from 0 to n-1
    for (int k=0; k <= j; k++)                           // loop from 0 to j
      total += arr[j];
  return total;
}

/** Returns the sum of the prefix sums of given array. */
public static int example4(int[ ] arr) {
  int n = arr.length, prefix = 0, total = 0;
  for (int j=0; j < n; j++) {                            // loop from 0 to n-1
    prefix += arr[j];
    total += prefix;
  }
  return total;
}
```