Laboratory practice No. 2: Big O Notation

Juan S. Cárdenas Rodríguez
Universidad EAFIT
Medellín, Colombia
jscardenar@eafit.edu.co

David Plazas Escudero
Universidad EAFIT
Medellín, Colombia
dplazas@eafit.edu.co

September 5, 2017

1) ONLINE EXERCISES (CODINGBAT)

1.a. Array II

i.

```
public int[] zeroFront(int[] nums) {              // c0
    boolean [] used = new boolean [nums.length]; // c1
    int cont = 0;                                 // c2
    for (int i = 0; i < nums.length; i++) {       // c3 + n
      if(nums[i] == 0) {                          // c4 + n
        if (i != cont) {                          // c5 + n
          nums[i] = nums[cont];                   // c6 + n
          nums[cont] = 0;                         // c7 + n
        }
        cont++;                                   // c8 + n
      }
    }
    return nums;                                  // c9
  }
```

Therefore, `zeroFront` is $O(c_0 + c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + c_9 + 6n)$. Applying the sum and product properties, `zeroFont` is $O(n)$.

ii.

```
public int[] notAlone(int[] nums, int val) {      // c0
    for(int i = 1; i < nums.length-1; i++) {      // c1 + n
      if(nums[i] == val && nums[i-1] != val
        && nums[i+1] != val) {                    // c2 + n
        if (nums[i-1] > nums[i+1])                // c3 + n
          nums[i] = nums[i-1];                    // c4 + n
        else                                      // c5 + n
          nums[i] = nums[i+1];                    // c6 + n
```

```
        }
      }
      return nums;                                    // c7
    }
```

Therefore, `notAlone` is $O(c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + 6n)$. Applying the sum and product properties, `notAlone` is $O(n)$.

iii.

```
public boolean tripleUp(int[] nums) {                 // c0
  for (int i = 0; i < nums.length - 2; i++) {         // c1 + n-2
    if(nums[i] + 1 == nums[i+1] && nums[i]
     + 2 == nums[i+2]) return true;                   // c2
  }
  return false;                                       // c3
}
```

`tripleUp` is $O(c_0 + c_1 + c_2 + c_3 + n - 2)$. When we apply the product and sum properties, `tripleUp` is $O(n)$.

iv.

```
public int[] tenRun(int[] nums) {                     // c0
  int tempMult = 0;                                   // c1
  boolean used = false;                               // c2
  for(int i = 0; i < nums.length; i++) {              // c3 + n
    if (nums[i] % 10 == 0) {                          // c4 + n
      used = true;                                    // c5 + n
      tempMult = nums[i];                             // c6 + n
    }                                                 // c7 + n
    if(used)
      nums[i] = tempMult;
  }
  return nums;
}
```

v.

```
public int[] shiftLeft(int[] nums) {
  int [] mod = new int[nums.length];
  if (nums.length==1) return nums;
  for (int i=1; i<nums.length; i++) {
    mod[nums.length-1]=nums[0];
    mod[i-1]=nums[i];
  }
  return mod;
}
```

Professor Mauricio Toro Bermúdez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

1.b. Array III

i.

```
public int[] seriesUp(int n) {
  int no = n*(n+1)/2;
  int [] nums = new int [no];
  int a = 0;
  for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
      nums[a] = j;
      a++;
    }
  }
  return nums;
}
```

ii.

```
public int countClumps(int[] nums) {
  int c = 0;
  for (int i = 0; i < nums.length-1; i++) {
    if (nums[i] == nums[i+1]) {
      for (int j = i; j < nums.length; j++) {
        if (nums[j] != nums[i]) {
          i = j;
          c++;
        }
        if (c == 0 && j == nums.length-1) {
          c++;
        }
      }
    }
  }
  return c;
}
```

iii.

```
public boolean linearIn(int[] outer, int[] inner) {
  int j = 0;
  int c = 0;
  if (inner.length == 0) {
    return true;
  }
  for (int i = 0; i < outer.length; i++) {
    if (inner[j] == outer[i]) {
      j++;
      if (j==inner.length) {
        return true;
```

Professor Mauricio Toro Bermúdez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

```
        }
      }
    }
    return false;
  }
```

iv.

```
  public int[] fix45(int[] nums) {
    boolean [] arr = new boolean[nums.length];
    for (int i = 0; i < nums.length-1; i++) {
      if (nums[i] == 4 && nums[i+1] == 5) {
        arr[i+1] = true;
      } else  if (nums[i] == 4 && nums[i+1] != 5) {
        for (int j = 0; j < nums.length; j++) {
          if (nums[j] == 5 && arr[j] == false) {
            nums[j] = nums[i+1];
            nums[i+1] = 5;
            arr[i+1] = true;
            break;
          }
        }
      }
    }
    return nums;
  }
```

2) SIMULATION OF PROJECT PRESENTATION QUESTIONS

2.a. ArrayMax

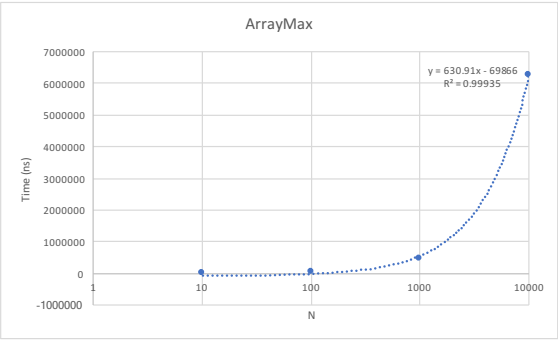

Figure 1: Time vs. N for ArrayMax

N	Time (ns)
10	6000
100	27000
1000	346000
10000	6717000

Table 1: ArrayMax's data.

3) EXAM SIMULATION

i. `start` $+ 1$, `nums`, `target`

ii. a) $T(n) = T(n/2) + C$

iii. $n - a, a, b, c$
res, solucionar$(n - b, a, b, c)$+1
res, solucionar$(n - c, a, b, c)$+1

iv. e) La suma de los elementos de a y es $O(n)$.

Professor Mauricio Toro Bermúdez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

References

Professor Mauricio Toro Bermúdez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co