ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 1: Recursion

Alejandro Salazar
Universidad Eafit
Medellín, Colombia
asalazara1@eafit.edu.co

Andrés Grimaldos
Universidad Eafit
Medellín, Colombia
agrimaldoe@eafit.edu.co

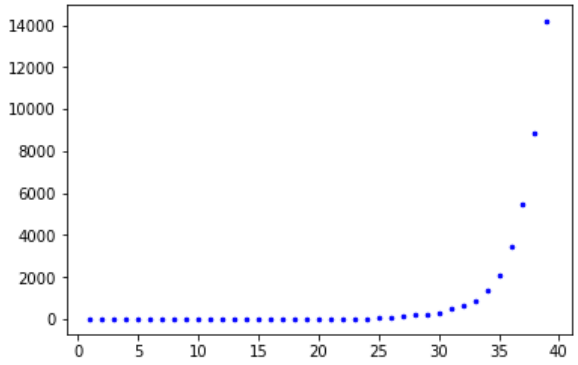
3) Practice for final project defense presentation

3.1

$$T(n) = T(n-1) + T(n-1) + C_1$$
$$T(n) = C_1 * 2^{n-1} + C_1 * (2^n - 1)$$
$$T(n) = C_1 * 2^{n-1} + C_1 * 2^n - C_1$$
$$O(n) = C_1 2^n$$
$$O(n) = 2^n$$

Addition's Rule
Product's Rule

3.2



n: 1 time: 0.00341796875
n: 2 time: 0.004150390625
n: 3 time: 0.003173828125
n: 4 time: 0.002685546875
n: 5 time: 0.0029296875
n: 6 time: 0.003662109375
n: 7 time: 0.0048828125
n: 8 time: 0.00732421875
n: 9 time: 0.01123046875
n: 10 time: 0.014892578125
n: 11 time: 0.022705078125
n: 12 time: 0.123046875
n: 13 time: 0.056884765625
n: 14 time: 0.087646484375
n: 15 time: 0.148193359375
n: 16 time: 0.221435546875
n: 17 time: 0.4091796875
n: 18 time: 0.62255859375
n: 19 time: 0.96875
n: 20 time: 1.482666015625

n: 21 time: 2.4755859375
n: 22 time: 3.8916015625
n: 23 time: 7.714111328125
n: 24 time: 10.956298828125
n: 25 time: 87.497314453125
n: 26 time: 95.17333984375
n: 27 time: 114.719482421875
n: 28 time: 202.936279296875
n: 29 time: 223.237060546875
n: 30 time: 311.22314453125
n: 31 time: 477.7900390625
n: 32 time: 625.92724609375
n: 33 time: 889.9765625
n: 34 time: 1367.695068359375
n: 35 time: 2114.3583984375
n: 36 time: 3428.544921875
n: 37 time: 5489.5517578125
n: 38 time: 8853.205322265625
n: 39 time: 14187.599853515625

Based on the graphic and the data obtained we estimate that for 50 elements the program is going to spend around 14.336 seconds (data obtained in milliseconds)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3.3 Considering the time the program spends and the way it acts depending on the number of centimeters that the rectangle has, we have concluded that it will not be useful for Puerto Antioquia because of the extremely high measures we can obtain from the containers

3.4 What the program GroupSum5 is that it receives an array and a target as parameters and then it returns if there is possible or not to find a combination of numbers in the array which's summation gives the target as result. What makes this program special is that every multiple of 5 in the array is included in the summation and that if there's a 1 next to a multiple of five (being the multiple of 5 first than the 1 in the array) that 1 will not be included in the summation.

3.5 Recursion 1

1. SumDigits:

$$T(n) = \begin{cases} C_1, & n = 0 \\ T(n-1) + C_2, & n > 0 \end{cases}$$

$$T(n) = C_2 n + C_1$$

$$O(n) = C_2 n$$

$$O(n) = n$$

2. Array6

$$T(m) = \begin{cases} C_1, & m = 0 \\ T(m-1) + C_2, & m > 0 \end{cases}$$

$$T(n) = C_2 m + C_1$$

$$O(n) = C_2 m$$

$$O(m) = m$$

3. BunnyEars

$$T(l) = \begin{cases} C, l = 0 \\ T(l-1) + C_2, l, \\ T(l-1) + C_3 \end{cases}$$

$$T(l) = \frac{C_2}{2} l + \frac{C_3}{2} l + C_1$$

$$T(l) = l\left(\frac{C_2 + C_3}{2}\right) + C_1$$

$$O(l) = l\left(\frac{C_2 + C_3}{2}\right)$$

$$O(l) = l$$

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4. Triangles

$$T(k) = \begin{cases} C_1, & k = 0 \\ T(k-1) + C_2, & k > 0 \end{cases}$$

$$T(k) = C_2 k + C_1$$

$$O(k) = C_2 k$$

$$O(k) = k$$

5. BunnyEars

$$T(l) = \begin{cases} C_1, & l = 0 \\ T(l-1) + C_2, & l > 0 \end{cases}$$

$$T(l) = C_2 l + C_1$$

$$O(l) = C_2 l$$

$$O(l) = l$$

Recursion 2

1. GroupNoAdj:

$$T(m) = \begin{cases} C_1, & m = 0 \\ 2T(m-1) + C_2, & m > 0 \end{cases}$$

$$T(m) = C_1 2^m + C_2(2^m - 1)$$

$$O(m) = C_1 2^m$$

$$O(m) = 2^m$$

2. SplitArray

$$T(m) = \begin{cases} C_1, & m = 0 \\ 2T(m-1) + C_2, & m > 0 \end{cases}$$
$$T(m) = C_1 2^m + C_2(2^m - 1)$$

$$O(m) = C_1 2^m$$

$$O(m) = 2^m$$

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3. GroupSum6

$$T(m) = \begin{cases} C_1, & m = 0 \\ 2T(m-1) + C_2, & m > 0 \end{cases}$$

$$T(m) = C_1 2^m + C_2(2^m - 1)$$

$$O(m) = C_1 2^m$$

$$O(m) = 2^m$$

4. GroupSum5

$$T(m) = \begin{cases} C_1, & m = 0 \\ 2T(m-1) + C_2, & m > 0 \end{cases}$$

$$T(m) = C_1 2^m + C_2(2^m - 1)$$

$$O(m) = C_1 2^m$$

$$O(m) = 2^m$$

5. GroupSumClump

$$T(m) = \begin{cases} C_1, & m = 0 \\ 2T(m-1) + C_2, & m > 0 \end{cases}$$

$$T(m) = C_1 2^m + C_2(2^m - 1)$$

$$O(m) = C_1 2^m$$

$$O(m) = 2^m$$

3.6
'n' is the number of digits that the number has
'm' is the difference between the number of elements from the array and the variable start
'k' is the number of rows of the triangle
'l' is the number of bunnies that are there

4) Practice for midterms

4.1 start + 1, nums, target
4.2 a

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4.3.1
Line 4: int res = solucionar(n-a, a, b, c) + 1;
4.3.2
Line 5: res = Math.max(res, solucionar(n-b, a, b ,c));
4.3.3
Line 6: res = Math.max(res, solucionar(n-c, a, b, c);
4.4 **e**
4.5.1
Line 2: if(n <= 2) return n;
Line 3: return formas(n-1) +
Line 4: formas(n-2);
4.5.2 b
4.6.1
Line 10: return sumaAux(n, i+2);
4.6.2
Line 12: return (n.charAt(i) - '0') + sumaAux(n, i+1);
4.7.1
Line 9: return comb(S, i+1, t)
4.7.2
Line 10: comb(S, i+1, t – S[i]);
4.8.1
Line 9: return 0;
4.8.2
Line 13: suma = ni + nj;
4.9 c
4.10 b
4.11.1
Line 4: return lucas(n-1) + lucas(n-2);
4.11.2 c
4.12.1
Line 13: return sat;
4.12.2
Line 17: sat += Math.max(fi, fj);
4.12.3
Line 18: return sat;

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473