Laboratory practice No. 01: Recursion

Miguel Angel Correa Manrique
Universidad Eafit
Medellín, Colombia
macorream@eafit.edu.co

Pablo Buitrago Jaramillo
Universidad Eafit
Medellín, Colombia
pbuitragoj@eafit.edu.co

3)Practice for final project defense presentation

3.1

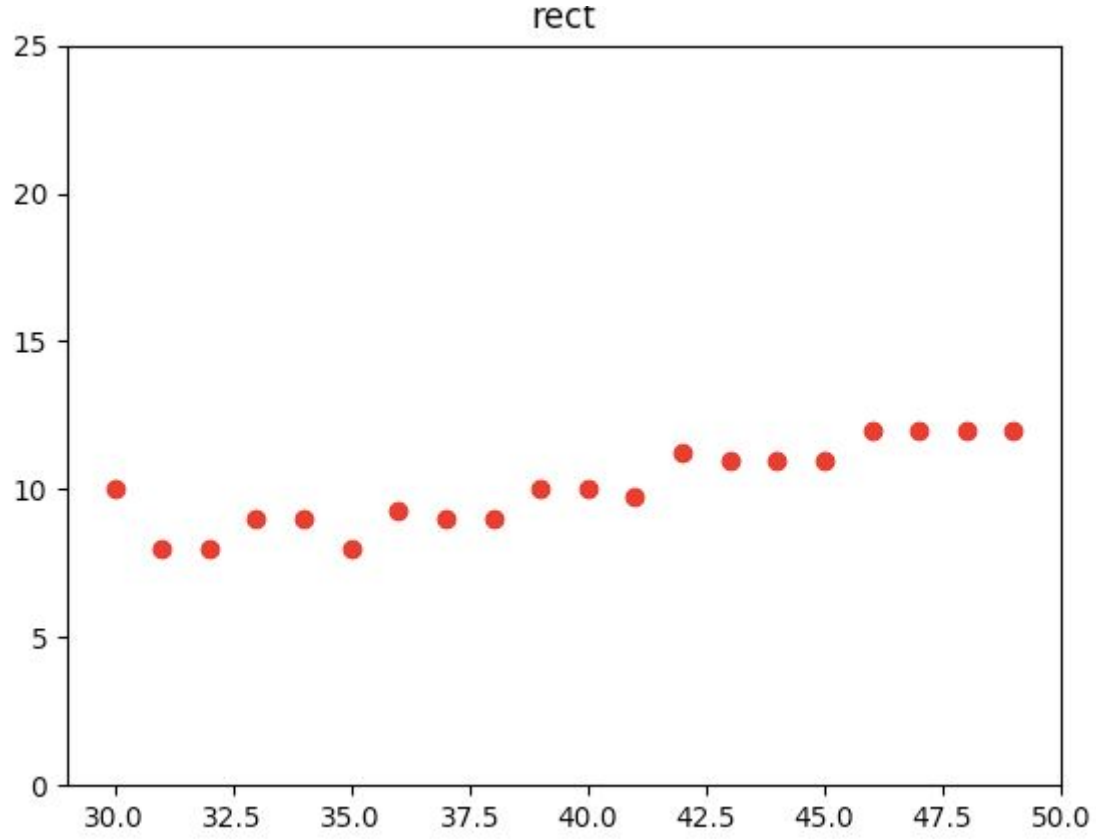
```
1 def rect(n):
2    '''
3    Return the number of ways how one can organize 1x2
4    rectangles in a 2xn rectangle.
5    '''
6    if n == 1 or n == 2: //c_1 --> O(1)
7        return n        // c_2 --> O(1)
8
9    arr = [0]           //c_3 --> O(1)
10
11    arr.append(1)       //c_4 --> O(1)
12    arr.append(2)       //c_5 --> O(1)
13
14    for i in range(3, n+1): //c_6 * n+1 --> O(n+1)
15        arr.append(arr[i-1] + arr[i-2]) //c_7 + c_8* n → O(n)
16
17    return arr[n] c_9 --> O(1)
```

This is not the most mathematical rigorous way on finding the Big O expression (it is going to be used down below for the rest of exercises), but for the purpose needed it, it is actually a good manner of representation. Therefore, since the fastest growing term is O(n+1), one takes out the coefficient, leaving us with linear time, meaning O(n).

Also obvious O(n) because it uses a for loop to iterate over n-elements.

3.2 (The Code used for this exercise is in a file called “Exercise3Code.py”)

Our program plots the time required to run the exercise 1.2 for the numbers from 30 to 50; As shown in the graph below, it is a linear model, so we estimate the spent time for n = 50 and it oscillates in the range 11.60*10^-6 < s < 12*10^-6 seconds



3.3 Considering a O(n) time complexity and the computer power that is possible to obtain nowadays, including Moore's Law which tells us that processing power for computers will double every two years, we conclude that it is viable to use this algorithm in Puerto Antioquia during the year 2020 (mostly because the fact that it is a linear model).

3.4 Seeing that the reader already knows the proposed algorithm to develop, the main focus will be how the implementation works, the function is going through the entire vector, finding all the possible subsets to match the target (can't avoid multiples of 5), if there is a five and then a 1, the 1 does not count as part of the subset; it is defined that if a number is "added" to the subset it is going to be subtracted from target; meaning that if there is a match, the target's value must be 0.

The algorithm will stop when the variable access operator is bigger than the actual size of the vector, evaluating if the target match was successful or not, 0 or another value respectively.

3.5

A. All Star

```
11 string allStar(string str, size_t n){
```

```
12      if (n == str.size()-1){ //c_1 → O(1)
13      return string{str[n]}; //c_2 → O(1)
14
15      }
16
17      return string{str[n], '*'} + allStar(str, n+1); // c_3 + c_4  + c_5 * (n+1) → O(n+1)
18  }
```

- Time complexity O(n+1) because of the fastest growing term rule, one takes out the coefficients and constants, leaving us with O(n); also it was defined n+1 because when algorithm when is initialized it wastes time on its first execution, having then the amount of time required to go through the entire array, being n.

B. Array220

```
27 bool array220(vector<int> nums, int index){
28      if(index == nums.size()-1 || nums.size() == 0) return false; //c_1 + c_2 → O(1)
29
30      if(nums[index]*10 == nums[index+1]) return true;  //c_1 + c_2 + c_3 → O(1)
31
32      else return false || array220(nums, index+1);  //c_4 + c_5 * n+1 → O(n+1)
33
34 }
```

- Because of the fastest growing term rule, time complexity is O(n+1) taking out the coefficients and constants, therefore O(n).

C. CountHi2

```
48 int countHi2(string str){
49      int index = str.find("ih"); // c_1 * n → O(n)
50      if(index == -1) return 0; // c_2 → O(1)
51      else{
52          if(index + 2 >= str.length()) return 1; // c_3 + c_4 → O(1)
53          else{
54              if(str[index+2] != 'x') return 1 + countHi2(str.substr(index++2)); // c_5 + c_6 + c_7
* k → O(k)
55              else return 0 + countHi2(str.substr(index+3)); //c_8 + c_9 * m → O(m)
56          }
57      }
58
59 }
```

- The fastest growing term is given by the function str.find which gives a time complexity of O(n), note that there are O(k) and O(m) time complexity for this algorithm, where k and m represents the length of the resulting substring.

D. Count7

```
2 public int count7(int n) {
3   if(n==0) return 0;  // c_1 → O(1)
4   else if(n%10==7) return 1+count7(n/10); // c_2 + c_3 * n+1 → O(n+1)
5   else return count7(n/10); // c_4 + c_5 * n+1 → O(n+1)
6 }
```

- On every case scenario, every function will lead the algorithm to run n+1 times, being "n" the length of the number given, and "1" the first execution function realices. Therefore we have a O(n) time complexity.

E. Count8

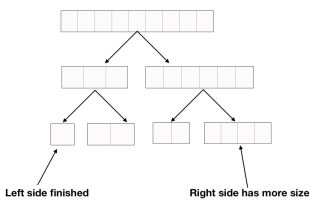
```
8 public int count8(int n) {
9   if(n==0) return 0; //c_1 → O(1)
10   if(n%100==88) return 2+count8(n/10); //c_2 + c_3 + c_4 * n+1 → O(n+1)
11   else if(n%10==8) return 1+count8(n/10); //c_5 + c_6 + c_7 * n+1 → O(n+1)
12   else return count8(n/10); c_8 * n+1 → O(n+1)
13 }
```

- Same as the one up, O(n) time complexity, just changes the value to search, but it goes through every position of the integer.

F. GroupSum6

```
8 bool groupSum6(size_t start, vector<int> nums, int target){
9       if(nums.size()-1 < start){ //c_1 → O(1)
10              if(target == 0) return true; //c_2  → O(1)
11              else return false; //c_3 → O(1)
12
13       }
14
15       if(nums[start] == 6) return groupSum6(start+1, nums, target-nums[start]); //c_4 + c_5 * (n+1)
→ O(n+1)
16       return groupSum6(start+1, nums, target-nums[start]) || groupSum6(start+1, nums, target);
   //c_6 + c_7 * 2T(n-1)   → O(2^n)
17
18 }
```

- Fastest growing term rule selects $O(2^n)$ time complexity, this because of the fact that 2T(n-1) will recreate branches for the algorithm tree, an example as follows:



G. GroupSum5

```
21 bool groupSum5(size_t start, vector<int> nums, int target){
22
23        if(nums.size()-1 < start) return target == 0; //c_1 → O(1)
24
25
 26              if(nums[start]%5 == 0 && nums.size()-1 >=  start+1 && nums[start+1] == 1) return
groupSum5(start+2, nums, target-nums[start]); c_2 *  n/2 → O(n)
27
 28            else if (nums[start]%5 == 0) return groupSum5(start+1, nums, target-nums[start]); c_3 +
c_4→ O(n+1)
29
 30        return groupSum5(start+1, nums, target-nums[start]) || groupSum5(start+1, nums, target); c_5
+ c_6 * T(n-1) → O(2^n)
```

- Just as above, the time complexity is $O(2^n)$ by the rule of the fastest growing term.

H. groupNoAdj

```
2 public boolean groupNoAdj(int start, int[] nums, int target) {
3   return (start >= nums.length) ? (target == 0) : (groupNoAdj(start+2, nums, target-nums[start]) ||
groupNoAdj(start+1, nums, target)); } c_1 + c_2 + c_3 * 2T(n-1) → O(2^n)
```

- As explained, the reason of $O(2^n)$ is because of how the recursion tree works, leading then to the fastest growing term rule.

I. groupSumClump

```
 6 public boolean groupSumClump(int start, int[] nums, int target) {
 7   if(start >= nums.length) return (target == 0); // c_1 + c_2 → O(1)
 8   int sum = 0; //c_3 → O(1)
 9   int clump = start; // c_4 → O(1)
 10   while(clump < nums.length && nums[start] == nums[clump]){
 11     sum += nums[start]; //c_5 → O(1)
 12     clump++; //c_5 → O(1)
 13   } → O(n)
 14   return groupSumClump(clump, nums, target) || groupSumClump(clump, nums, target-sum);
2T(n-1)  → O(2^n)
15 }
```

- Same behavior, function tree generates a complexity of $O(2^n)$ defined as above.

J. splitArray

```
17 public boolean splitArray(int[] nums) {
18   return splitArrayAux(0, 0, nums, 0);
19 }
20 private boolean splitArrayAux(int start, int sum1, int[] nums, int sum2){
21   if(start >= nums.length) return (sum1 == sum2); //c_1 } c_2 → O(n)
22   return splitArrayAux(start+1, sum1+nums[start], nums, sum2) || splitArrayAux(start+1, sum1,
nums, sum2+nums[start]); c_1 + c_ 2 + 2T(n-1) → O(2^n)
23 }
```

- By fastest growing term rule $O(2^n)$, since the function is called 2T(n-1) times.

Note: Recurrence equations are given as follows

- For exercises on Recursion 1 T = T(n-1) + C, if n > 0 or T = C if n == 0
- For exercises on Recursion 2 T = 2T(n-1) + C, if n > 0 or T = C if n == 0

3.6

Since Wolfram was not needed, every variable could be described in the table as follows:

Variable	Meaning
n	Numbers of elements in an array.
c	Constant operations
m (On exercise CountHi2)	Number of elements substring applied
k (On exercise CountHi2)	Number of elements substring applied

4)Practice for midterms

4.1 (op) start+1, nums, target

4.2 a. T(n/2) + C

$$T(n) = \begin{cases} \Theta(1), & n = 1 \text{ (low = high)} \\ T\left(\frac{n}{2}\right) + \Theta(1), & \text{if } n > 1 \end{cases}$$

4.3

4.3.1 n-a, a, b, c

4.3.2 res, solucionar(n-b, a, b, c)+1

4.3.3 res, solucionar(n-c, a, b, c)+1

4.4 (op) e.

4.5

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. T(n−1)+T(n−2) + C

4.6

4.6.1 return sumaAux(n, i+2);

4.6.2 return (n.charAt(i) - '0') + sumaAux(n.substring(i+1), i);

4.7 (op)

4.7.1 return comb(S, i+1, t) ||

4.7.2 return comb(S, i+1, t-S[i]);

4.8

4.8.1 return 0;

4.8.2 return ni + nj;

4.9 (op) c

4.10 b

4.11

4.11.1 return lucas(n-1) + lucas(n-2);

4.11.2 c) T(n)=T(n−1)+T(n−2)+c, que es O(2^n***)***

4.12

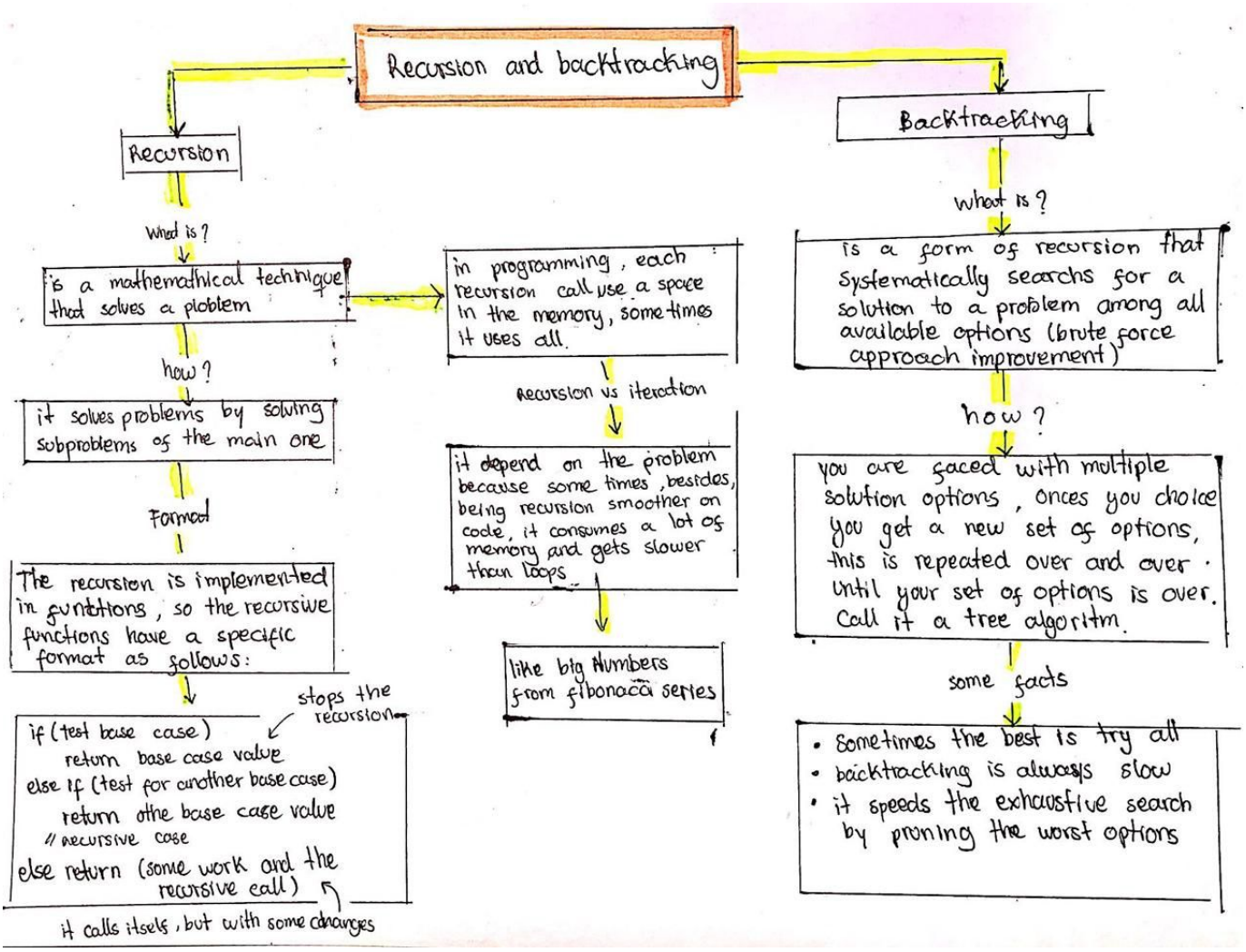
4.12.1 return sat;

4.12.2 sat += Math.max(fi, fj);

4.12.3 return sat;

5)Narasimha Karumanchi - Data structures and algorithms made easy in java. Recursion and Backtracking

Abstract:
Recursion is a very useful technique borrowed from mathematics that is used to solve tasks that can be defined in terms of similar subtasks. Any function that calls itself is called a recursive function, it solves a problem by calling a copy of itself to work on a smaller problem, this can result in many recursive calls so it is fundamental to ensure that the recursion stops when some flag event is reached. Below you will see about recursion and some other type of recursion that works like tree algorithms, pruning solution options, backtracking.



References:
1.Karumanchi, N. *Data structures and algorithms made easy in Java*. 2017.

6)Team work and gradual progress (optional)

6.1 Meeting minutes

Member	Date	Done	Doing	To do
Miguel	17/08/2019	Exercises 1.1 on C++ and 1.2 on Python using Dynamic Programming.	3 Recursion 1 2 Recursion 2 Codingbat exercises	Worksheet, points 3 and 4.
Miguel	18/08/2019	3 : Recursion 1 exercises	2 : Recursion 2 Codingbat	Worksheet, points 3 and 4.

		implemented on C++.	exercises	
Pablo	18/08/2019	2 : Recursion 1 3 : Recursion 2 Exercises on JAVA implemented.	Worksheet point 3.	Worksheet point 4.
Miguel	18/08/2019	2 : Recursion 2 exercises implemented on C++.	Worksheet point 3.	Worksheet point 4.
Pablo	18/08/2019	Point 3.2 solved	Points 3.3 and 3.4	Worksheet point 4.
Miguel	18/08/2019	Point 3.1 solved	Point 3.5 to 3.6	Worksheet point 4.
Pablo	18/08/2019	Points 3.3 and 3.4 solved	Worksheet point 4.	Optional Lecture
Pablo	18/08/2019	Most of the points on worksheet point 4 solved.	Worksheet point 4.	Optional Lecture
Miguel	18/08/2019	Point 3.5 to 3.6 solved.	Worksheet point 4.	Teamwork gradual process optional point.
Miguel	18/08/2019	Time complexity points on worksheet point 4 solved.		
Pablo	18/08/2019	All points from worksheet point 4 solved, except last one.	Optional Lecture.	
Miguel & Pablo	18/08/2019	Point left on worksheet point 4.	Optional Lecture Teamwork gradual process point.	
Pablo	18/08/2019	Optional Lecture.		
Miguel	18/08/2019	Teamwork		

		gradual process point.		

6.2 History of changes of the code

```
|\  Merge: be88455 c2b2700
| | Author: miguelmque <macorream@eafit.edu.co>
| | Date:   Sun Aug 18 23:09:56 2019 -0500
| |
| |     Merge remote-tracking branch 'origin'
| |
| * commit c2b270041a378a9ecded09f586caf4ffbe29aa0f
|                       |                   Author:                   pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 18:06:14 2019 -0500
| |
| |     Update and rename Excercise3Code.py to Exercise3Code.py
| |
| * commit 7e849ddc0c739dbde0cfe146f56d109a9277609d
|                       |                   Author:                   pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 17:55:55 2019 -0500
| |
| |     Create Excercise3Code.py
| |
| |     Cree el archivo y pegué el código correspondiente al punto 3.2
| |
* | commit be88455cb2f9a0ecc84e2913c12c2b53ffd83bcf
|/  Author: miguelmque <macorream@eafit.edu.co>
|   Date:   Sun Aug 18 23:09:14 2019 -0500
|
|       change on description
|
*   commit c6aafe69dc15d871d437f4e50fd36d7f230b3fc5
|\  Merge: cf8c66d 768638c
| | Author: miguelmque <macorream@eafit.edu.co>
| | Date:   Sun Aug 18 17:32:43 2019 -0500
| |
| |     Merge remote-tracking branch 'origin'
| |
| * commit 768638c715bb12390cff26ec22b0c21e14aec1c9
```

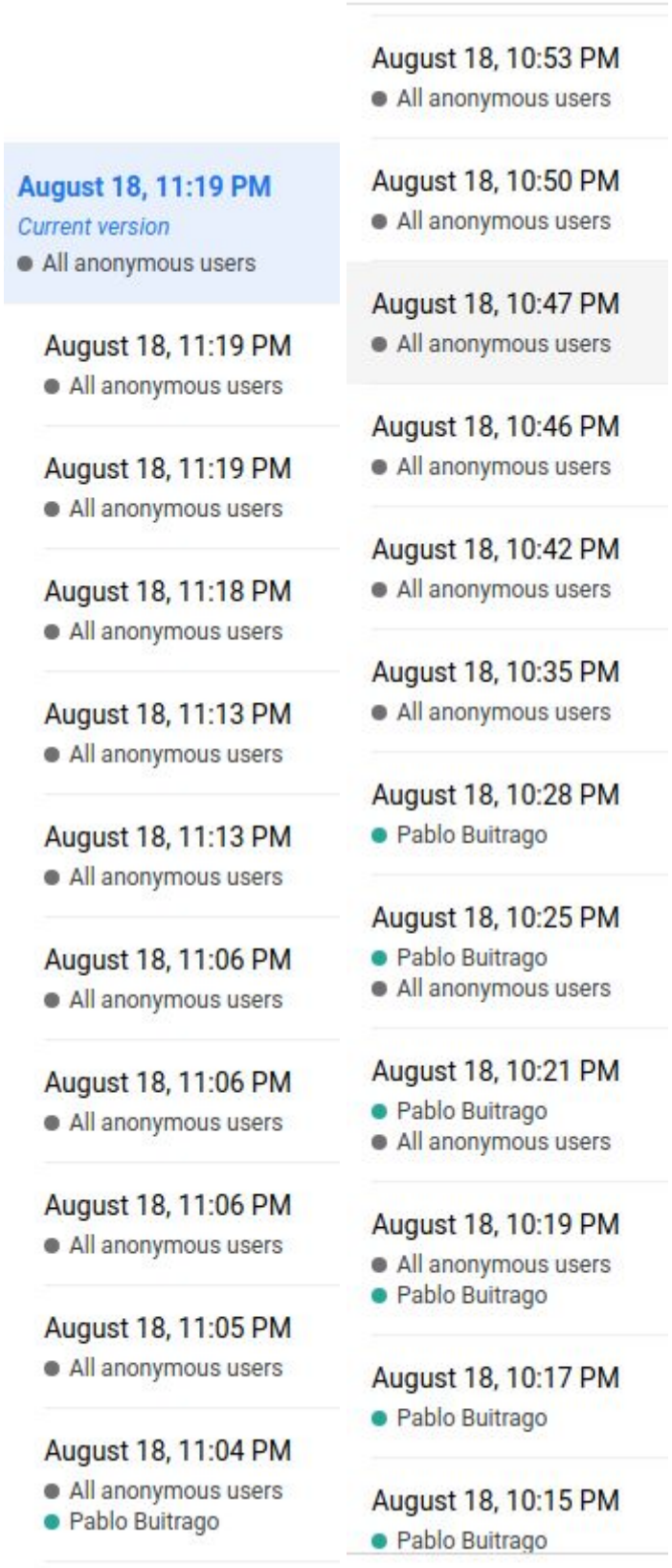
```
| | Author: pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 17:28:16 2019 -0500
| |
| |     Update Recursion1_part2.java
| |
* | commit cf8c66da60f2b5f1e64cede551facd1863812bb1
| | Author: miguelmque <macorream@eafit.edu.co>
| | Date:   Sun Aug 18 17:28:59 2019 -0500
| |
| |     added groupSum5 with vectors
| |
* |   commit a1afd850a7e989e793373aa313e2991e8a55f816
|\ \  Merge: fbc4708 4da6d27
| |/  Author: miguelmque <macorream@eafit.edu.co>
| |   Date:   Sun Aug 18 17:28:04 2019 -0500
| |
| |       Merge remote-tracking branch 'origin'
| |
| * commit 4da6d274bbeec98c7c98481fa9a5e2ee44f62a86
| | Author: pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 17:26:14 2019 -0500
| |
| |     Update Recursion2_part2.java
| |
| * commit 665c96317eaaea4fc6aeb7e9c75ea76b6244b038
| | Author: pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 16:48:30 2019 -0500
| |
| |     Create Recursion2_part2.java
| |
| * commit 08cf8ed62f7f04ceaea028523faf2edfb566fbe1
| | Author: pablo4buitrago
<52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 16:47:30 2019 -0500
| |
| |     Rename Recursion1.java to Recursion1_part2.java
| |
| * commit fc533201bf445d1354c269fa7a180856c3b233ce
```

```
| | Author: pablo4buitrago <52968548+pablo4buitrago@users.noreply.github.com>
| | Date:   Sun Aug 18 16:47:04 2019 -0500
| |
| |     Create Recursion1.java
| |
* | commit fbc47080798260bd965a0fb66c78111935a49ba4
|/  Author: miguelmque <macorream@eafit.edu.co>
|   Date:   Sun Aug 18 17:26:08 2019 -0500
|
|       exercises changed to work with vectors, pointers are scary but interesting
|
* commit 9d17db9a75b5c1c363be2f509bc25a29d740c0e8
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 16:28:48 2019 -0500
|
|     GroupSum 6 added
|
* commit cc3c398da128144c1db31dccf210e9485cee0a8a
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 15:47:13 2019 -0500
|
|     exercise countHi2 added
|
* commit 578c46551b61571e1a4bce6a772f8fde66bcc883
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 14:10:10 2019 -0500
|
|     java code removed, replaced with c++ code instead
|
* commit bef601ad20b1f3d3aa31f45e9b7af5e711a53aa5
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 13:13:17 2019 -0500
|
|     array220 exercise added
|
* commit adbc07d9bcf0bcd96ecd223244ba4ef9580b0971
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 11:41:35 2019 -0500
|
|     allStar exercise
|
```

```
*   commit 4141d4e9311765d88921d7412c2e2eed8d8dfe7d
|\  Merge: ae0d7d8 dd4dcbf
| | Author: miguelmque <macorream@eafit.edu.co>
| | Date:   Sun Aug 18 06:04:46 2019 -0500
| |
| |     Merge remote-tracking branch 'origin/master'
| |
| * commit dd4dcbff3d93a4b1a46fb4d734584c15b2ec91e1
| | Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>
| | Date:   Sun Aug 18 04:51:40 2019 -0500
| |
| |     Delete problem1_1
| |
| |     wrong file added, merged already, documentation added in .cpp
| |
* | commit ae0d7d8aa74c6aca22526c731b6e9d6df3a06946
|/  Author: miguelmque <macorream@eafit.edu.co>
|   Date:   Sun Aug 18 05:59:35 2019 -0500
|
|       exercise 1.2 added, documented and dynamic
|
* commit 591594a5f87ed95d0122f254b5dac771e5414983
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 04:49:38 2019 -0500
|
|     documentation added, memoized solution
|
*   commit 9af94cc013c018a9f729d48b6d8cdd972329d4fe
|\  Merge: 62b226c b352790
| | Author: miguelmque <macorream@eafit.edu.co>
| | Date:   Sun Aug 18 04:47:49 2019 -0500
| |
| |     Merge remote-tracking branch 'origin'
| |
| * commit b3527906e2e4f31146f7c1834b65645466e38760
| |
| * commit b3527906e2e4f31146f7c1834b65645466e38760
| | Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>
| | Date:   Sun Aug 18 04:24:34 2019 -0500
| |
| |     Rename lcs.cpp to problem1_1.cpp
| |
```

```
| |     renamed file, solved memoized
| |
* | commit 62b226c03d6e835780ca479efbbf8c47d8c700fa
|/  Author: miguelmque <macorream@eafit.edu.co>
|   Date:   Sun Aug 18 04:45:09 2019 -0500
|
|       documentation added
|
* commit 9a9efc05bf57def14459b52c2dc68c5a807945c3
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 04:21:12 2019 -0500
|
|     memoized solution problem solved
|
* commit 80b3233309509ad9d9bac8db92751d5dae858a4c
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 01:43:52 2019 -0500
|
|     global variables added again
|
* commit 629d26cc750f6fb3646808cb61558b02adf9e0af
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 01:41:36 2019 -0500
|
|     lcs code cleaned
|
* commit ffd823212089c19be744513b9e108ed0f88811ea
| Author: miguelmque <macorream@eafit.edu.co>
| Date:   Sun Aug 18 01:36:15 2019 -0500
|
|      LCS problem solved, C++, requires attention, matrix declared as default has
to be dynamic
```

6.3 History of changes of the report



August 18, 10:15 PM
- All anonymous users

August 18, 10:15 PM
- Pablo Buitrago

August 18, 10:15 PM
- Pablo Buitrago

August 18, 10:14 PM
- All anonymous users
- Pablo Buitrago

August 18, 10:11 PM
- Pablo Buitrago

August 18, 10:10 PM
- All anonymous users

August 18, 10:10 PM
- All anonymous users

August 18, 10:08 PM
- Pablo Buitrago

August 18, 10:04 PM
- All anonymous users
- Pablo Buitrago

August 18, 10:02 PM
- All anonymous users

August 18, 10:02 PM
- Pablo Buitrago
- All anonymous users

August 18, 10:01 PM
- All

August 18, 9:59 PM
- All anonymous users

August 18, 9:59 PM
- Pablo Buitrago
- All anonymous users

August 18, 9:58 PM
- All anonymous users

August 18, 9:57 PM
- All anonymous users

August 18, 9:56 PM
- Pablo Buitrago

August 18, 9:54 PM
- All anonymous users
- Pablo Buitrago

August 18, 9:52 PM
- All anonymous users
- Pablo Buitrago

August 18, 9:49 PM
- All anonymous users

August 18, 9:49 PM
- Pablo Buitrago

August 18, 9:47 PM
- All anonymous users

August 18, 9:45 PM
- Pablo Buitrago

August 18, 9:44 PM
- All anonymous users

August 18, 9:16 PM Pablo Buitrago All anonymous users	August 18, 9:16 PM Pablo Buitrago All anonymous users
August 18, 9:14 PM All anonymous users	August 18, 9:14 PM All anonymous users
August 18, 9:13 PM All anonymous users	August 18, 9:13 PM All anonymous users
August 18, 9:12 PM All anonymous users	August 18, 9:12 PM All anonymous users
August 18, 9:10 PM All anonymous users	August 18, 9:10 PM All anonymous users
August 18, 9:08 PM All anonymous users	August 18, 9:08 PM All anonymous users
August 18, 9:07 PM All anonymous users	August 18, 9:07 PM All anonymous users
August 18, 9:06 PM All anonymous users	August 18, 9:06 PM All anonymous users
August 18, 9:06 PM All anonymous users	August 18, 9:06 PM All anonymous users
August 18, 9:05 PM All anonymous users	August 18, 9:05 PM All anonymous users
August 18, 9:04 PM All anonymous users	August 18, 9:04 PM All anonymous users
August 18, 9:02 PM All anonymous users	August 18, 9:02 PM All anonymous users

August 18, 9:02 PM
- All anonymous users

August 18, 9:00 PM
- All anonymous users

August 18, 8:58 PM
- All anonymous users

August 18, 8:58 PM
- All anonymous users

August 18, 8:56 PM
- All anonymous users

August 18, 8:55 PM
- All anonymous users

August 18, 8:53 PM
- All anonymous users

August 18, 8:52 PM
- All anonymous users

August 18, 8:50 PM
- All anonymous users

August 18, 8:48 PM
- All anonymous users

August 18, 8:48 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:46 PM
- Pablo Buitrago
- All anonymous users

August 18, 8:42 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:38 PM
- All anonymous users

August 18, 8:37 PM
- All anonymous users

August 18, 8:36 PM
- All anonymous users

August 18, 8:36 PM
- All anonymous users

August 18, 8:35 PM
- All anonymous users

August 18, 8:33 PM
- All anonymous users

August 18, 8:32 PM
- All anonymous users

August 18, 8:31 PM
- All anonymous users

August 18, 8:31 PM
- Pablo Buitrago
- All anonymous users

August 18, 8:29 PM
- Pablo Buitrago

August 18, 8:28 PM
- Pablo Buitrago

August 18, 8:42 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:38 PM
- All anonymous users

August 18, 8:37 PM
- All anonymous users

August 18, 8:36 PM
- All anonymous users

August 18, 8:36 PM
- All anonymous users

August 18, 8:35 PM
- All anonymous users

August 18, 8:33 PM
- All anonymous users

August 18, 8:32 PM
- All anonymous users

August 18, 8:31 PM
- All anonymous users

August 18, 8:31 PM
- Pablo Buitrago
- All anonymous users

August 18, 8:29 PM
- Pablo Buitrago

August 18, 8:28 PM
- Pablo Buitrago

August 18, 8:26 PM
- All anonymous users

August 18, 8:26 PM
- All anonymous users

August 18, 8:24 PM
- All anonymous users

August 18, 8:23 PM
- All anonymous users

August 18, 8:22 PM
- All anonymous users

August 18, 8:20 PM
- All anonymous users

August 18, 8:19 PM
- Pablo Buitrago

August 18, 8:12 PM
- All anonymous users

August 18, 8:11 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:09 PM
- All anonymous users

August 18, 8:08 PM
- All anonymous users

August 18, 8:08 PM
- Pablo Buitrago
- All anonymous users

August 18, 8:05 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:03 PM
- Pablo Buitrago

August 18, 8:02 PM
- All anonymous users
- Pablo Buitrago

August 18, 8:01 PM
- All anonymous users

August 18, 8:01 PM
- Pablo Buitrago
- All anonymous users

August 18, 8:00 PM
- All anonymous users
- Pablo Buitrago

August 18, 7:59 PM
- Pablo Buitrago
- All anonymous users

August 18, 7:54 PM
- All anonymous users
- Pablo Buitrago

August 18, 7:53 PM
- All anonymous users

August 18, 7:52 PM
- Pablo Buitrago
- All anonymous users

August 18, 7:49 PM
- All anonymous users

August 18, 7:47 PM
- All anonymous users

August 18, 7:46 PM
- All anonymous users

August 18, 7:44 PM
- All anonymous users

August 18, 7:43 PM
- All anonymous users

August 18, 7:42 PM
- All anonymous users

August 18, 7:41 PM
- All anonymous users

August 18, 7:40 PM
- All anonymous users

August 18, 7:40 PM
- All anonymous users

August 18, 7:38 PM
- All anonymous users

August 18, 7:37 PM
- All anonymous users

August 18, 7:35 PM
- All anonymous users

August 18, 7:33 PM
- All anonymous users

August 18, 7:15 PM

August 18, 6:46 PM
All anonymous users

August 18, 6:45 PM
All anonymous users

August 18, 6:43 PM
Pablo Buitrago

August 18, 6:42 PM
Pablo Buitrago
All anonymous users

August 18, 6:41 PM
Pablo Buitrago

August 18, 6:40 PM
Pablo Buitrago

August 18, 6:27 PM
Pablo Buitrago

August 18, 6:26 PM
Pablo Buitrago

August 18, 6:26 PM
All anonymous users
Pablo Buitrago

August 18, 6:25 PM
Pablo Buitrago

August 18, 6:24 PM
Pablo Buitrago
All anonymous users

August 18, 6:23 PM
Pablo Buitrago

August 18, 6:46 PM
All anonymous users

August 18, 6:45 PM
All anonymous users

August 18, 6:43 PM
Pablo Buitrago

August 18, 6:42 PM
Pablo Buitrago
All anonymous users

August 18, 6:41 PM
Pablo Buitrago

August 18, 6:40 PM
Pablo Buitrago

August 18, 6:27 PM
Pablo Buitrago

August 18, 6:26 PM
Pablo Buitrago

August 18, 6:26 PM
All anonymous users
Pablo Buitrago

August 18, 6:25 PM
Pablo Buitrago

August 18, 6:24 PM
Pablo Buitrago
All anonymous users

August 18, 6:23 PM
Pablo Buitrago

August 18, 6:18 PM
- All anonymous users
- Pablo Buitrago

August 18, 6:09 PM
- Pablo Buitrago

August 18, 6:07 PM
- Pablo Buitrago

August 18, 6:06 PM
- Pablo Buitrago

August 18, 6:05 PM
- Pablo Buitrago

August 18, 6:05 PM
- Pablo Buitrago

August 18, 6:03 PM
- Pablo Buitrago

August 18, 6:02 PM
- Pablo Buitrago

August 18, 6:02 PM
- Pablo Buitrago

August 18, 6:02 PM
- Pablo Buitrago

August 18, 6:00 PM
- Pablo Buitrago

August 18, 6:00 PM
- Pablo Buitrago