Final Project

Selection Sort

Jorge Barrientos

Eduardo Arredondo

April 3, 2019

ISW 0910

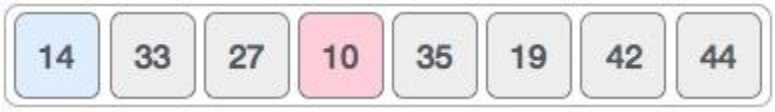
Content

Selection Sort

Selection sort is an algorithm that sorts an array by analyzing it recurrently and placing the numbers in order. This algorithm uses two sub arrays, the sorted array and the unsorted array, it´s to make the process faster instead of analyzing the whole array again and again.

Here is an example of how it works, first of all we need to have an unsorted array like this:

Next, the algorithm analyzes the whole array, then it identifies the lowest number in it. In this case is 10 is the lowest.

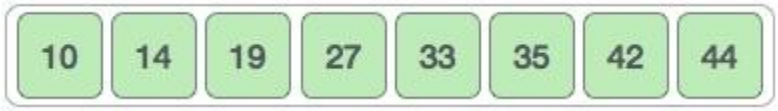


Once the lowest is identified it has to be placed in the first position, here, 14 is in the first position so 14 and 10 are going to be switched.

For the second position the algorithm just analyzes from 33 until 44 and repeats the same process as before it identifies the lowest and places it into the next position.

The next image shows how the array looks after doing the process for four times. Here the numbers in green are in the sorted sub-array and the other numbers are in the unsorted sub-array.

The final result should look like this, as you can see the numbers are in order and all of them are in the sorted array.



Code

In the image bellow is one example of selection sort, the code is commented with everything explained.

```
def sort(numbers): #here the function is declared and it will recieve the parameter of numbers

    for i in range(5): #here bebins a loop that is going to stop at 5 beginning from 0
        mainposition = i #here de variable i is the min position
        for j in range(i,6): #here starts our second loop witch starts from i witch is the min position and stops at 6
            if numbers[j] < numbers[mainposition]: #
                mainposition = j
# if numbers in j are lower  than numbers in mainposition then change the min position to j
        temp = numbers[i]
        numbers[i] = numbers[mainposition]
        numbers[mainposition] = temp
#here we swaping i to mainposition
numbers = [5,6,4,3,7,8]
sort(numbers)

def test_check(): #here we create out test function it must be called test_nameofthefunction
 assert numbers == [3,4,5,6,7,8] #here we placed the way the array should be and compare it with our variable numbers
```

Pytest

Is a framework used to test python code it works for both small test and complex functional test.

The code previously explained was tested with Pytest, to do that first you need to list in your path variable python and when installing you must install the pip pack.

Once it is installed, the cmd is opened or executed and go to your file where python is installed get into the scrips folder and type pip install pytest and just hit enter and it will show you something like this.

```
  Using cached https://files.pythonhosted.org/packages/7e/16/83b2a35c427b838df9836c9e7e4ae6dfbcbdea643
b44652f693b1c57d70/pytest-4.4.0-py2.py3-none-any.whl
Requirement already satisfied: py>=1.5.0 in c:\program files (x86)\python37-32\lib\site-packages (from
pytest) (1.8.0)
Requirement already satisfied: six>=1.10.0 in c:\users\jorge\appdata\roaming\python\python37\site-pack
ges (from pytest) (1.12.0)
Requirement already satisfied: atomicwrites>=1.0 in c:\program files (x86)\python37-32\lib\site-packag
s (from pytest) (1.3.0)
Requirement already satisfied: pluggy>=0.9 in c:\program files (x86)\python37-32\lib\site-packages (fr
m pytest) (0.9.0)
Requirement already satisfied: setuptools in c:\program files (x86)\python37-32\lib\site-packages (fro
 pytest) (40.8.0)
Requirement already satisfied: more-itertools>=4.0.0; python_version > "2.7" in c:\program files (x86)
python37-32\lib\site-packages (from pytest) (7.0.0)
Requirement already satisfied: colorama; sys_platform == "win32" in c:\users\jorge\appdata\roaming\pyt
on\python37\site-packages (from pytest) (0.4.1)
Requirement already satisfied: attrs>=17.4.0 in c:\program files (x86)\python37-32\lib\site-packages (
rom pytest) (19.1.0)
Installing collected packages: pytest
Successfully installed pytest-4.4.0
```

Remember, it must say, successfully installed!

After installing pytest in our computer the next step is to go to the folder que your phyton code is and just type into your cmd : pytest nameofyourfile.py and just hit enter, if there is any error it will shoe you something like this.

```
C:\Users\Jorge\Downloads>pytest selectorsort.py
============================= test session starts =============================
platform win32 -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: C:\Users\Jorge\Downloads
collected 1 item

selectorsort.py F                                                        [100%]

================================== FAILURES ===================================
_________________________________ test_check __________________________________

    def test_check():
>    assert numbers == [3,4,4,6,7,8]
E    assert [3, 4, 5, 6, 7, 8] == [3, 4, 4, 6, 7, 8]
E      At index 2 diff: 5 != 4
E      Use -v to get the full diff

selectorsort.py:17: AssertionError
========================== 1 failed in 0.14 seconds ===========================
```

Here we can see that our error was in the array, I purposely change the five for a four and it threw the error, it says that at index 2 a five was expected and it got a 4.

Now we fix that error and run it again.

```
C:\Users\Jorge\Downloads>pytest selectorsort.py
============================= test session starts =============================
platform win32 -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: C:\Users\Jorge\Downloads
collected 1 item

selectorsort.py .                                                        [100%]

========================== 1 passed in 0.27 seconds ===========================
```

As we can see it tested one item and everything run well.

PyTest is a tool very easy to use, it tells you where the error is and it also tells you what is happening, why is that an error.

Code Coverage

Is a tool for measuring lines, blocks and arcs executed in your code, this is useful to know which lines are useless to have a cleaner and more understandable code.

Code coverage can be run also from terminal, first you need to install it with the next command; pip install coverage, hit enter and the installation should start right away.

To check the coverage of a file you need to go in your terminal to the folder where you have your file, in this case mine is in the folder Almacen, so you just need to type; coverage run nameofyourfile.py after running it won't do anything you need to check it with the next command; coverage report –m it will show you all the files run previously and its statistics like this.

```
C:\Users\Jorge\Desktop\almacen>coverage run selectorsort.py

C:\Users\Jorge\Desktop\almacen>coverage report -m
Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
selectorsort.py      13      1    92%   17
```

The report shows that my file is 92% covered.

To see a better report you can type the next command; coverage html. It will create a html file with more detailed information like this.

Coverage for **selectorsort.py** : 92%

13 statements | 12 run | 1 missing | 0 excluded

```
1  def sort(numbers):
2
3          for i in range(5):
4                  mainposition = i
5                  for j in range(i,6):
6                          if numbers[j] < numbers[mainposition]:
7                                  mainposition = j
8
9                  temp = numbers[i]
10                 numbers[i] = numbers[mainposition]
11                 numbers[mainposition] = temp
12
13 numbers = [5,6,4,3,7,8]
14 sort(numbers)
15
16 def test_check():
17  assert numbers == [3,4,4,6,7,8]
18
```

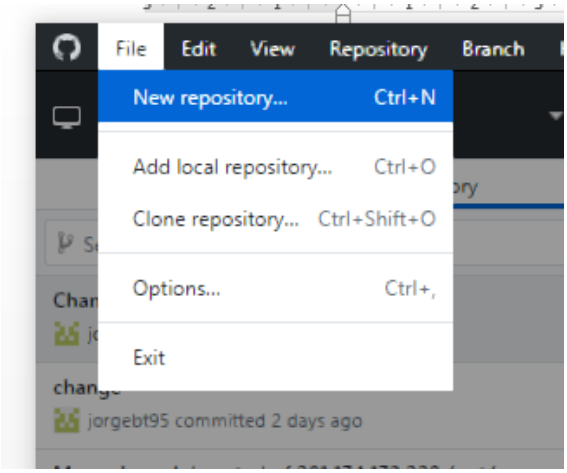
« index coverage.py v4.5.3, created at 2019-04-03 14:34

It was the first error from the first example. As we can see is a very useful tool and also very easy to use, it tells shows you with more precision where your error is and what didn't go well. So I highly recommend these two tools used in the project.

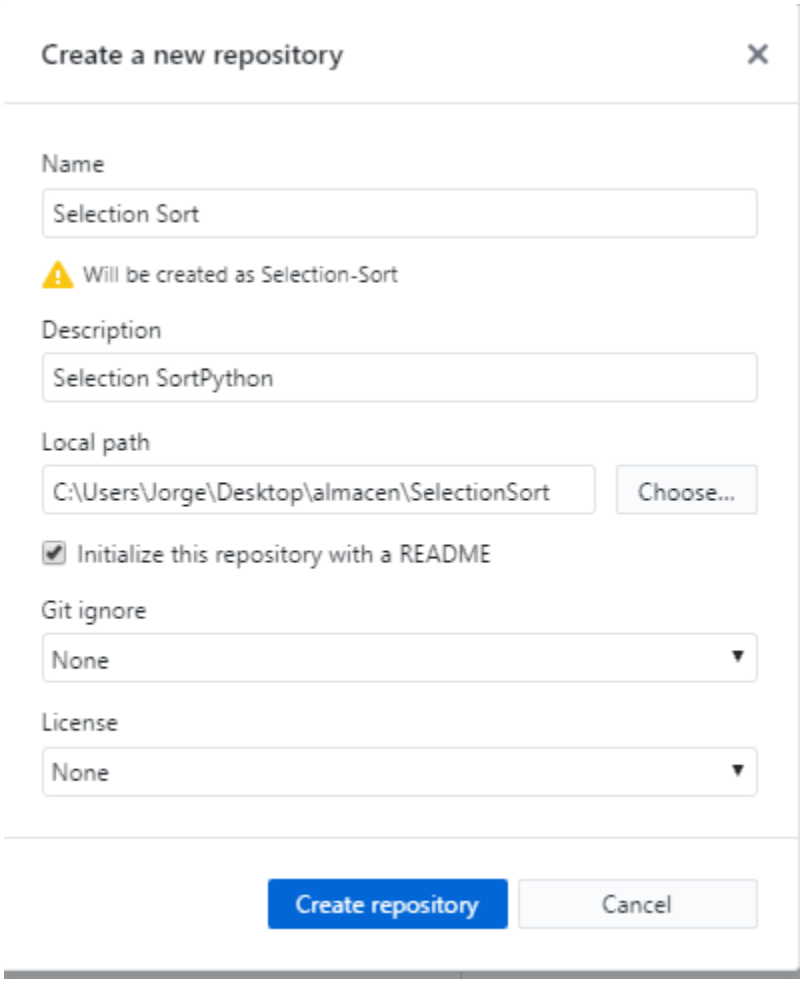
GitHub

Before trying to create a repository I´ll recommend to have downloaded the GitHub desktop app and also have an active account if not then click here.

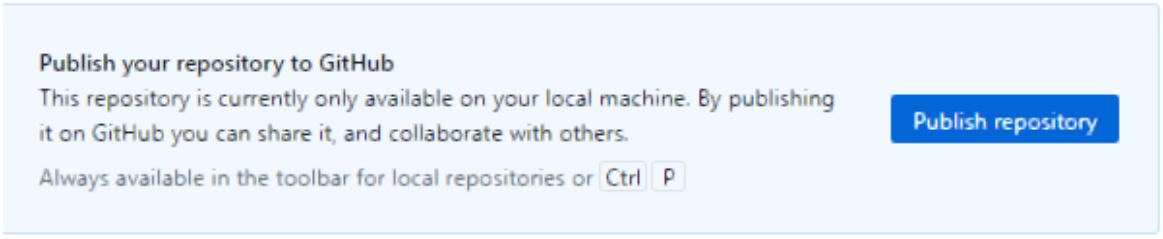
First you need to open the app click on file and then go to new repository.



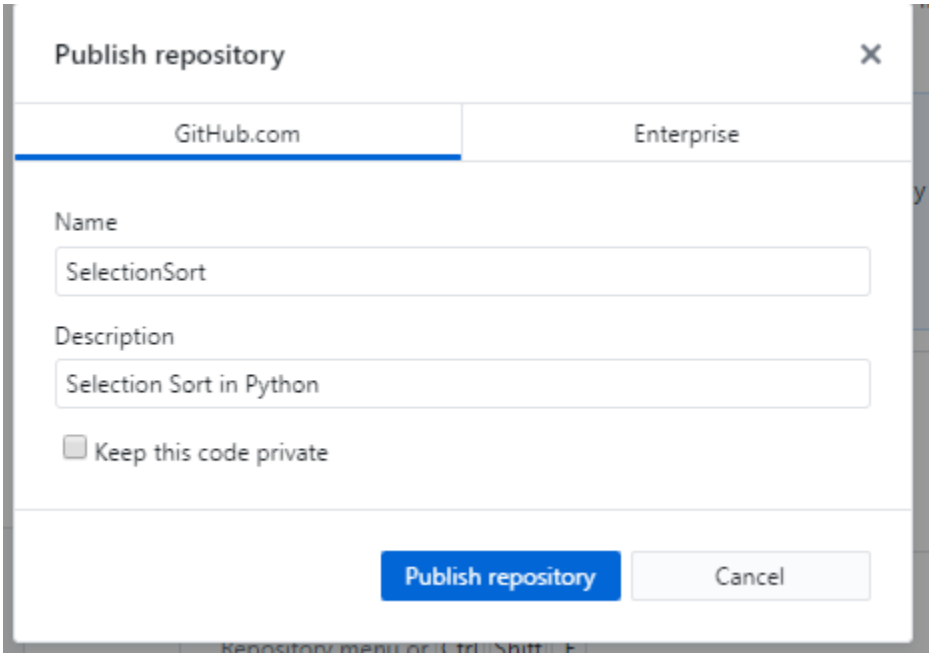
After clicking new repository it will show a window like this.



Fill the blanks, select where you want to make your repository and click Create repository. Then in your main view of the GitHub app will appear the next window.



Click on Publish repository and publish it into github.com and if you want everybody to see your code than make sure you leave the box unchecked where it says Keep this code private.



Click on publish repository and go to the GitHub webpage and check if the push was done successfully.

Here is the link of my code https://github.com/jorgebt95/SelectionSort , for further information you can contact us by email jorgebt95@gmail.com .