

Informe de práctica Nro. 01

Implementación y análisis de algoritmos básicos de ordenamiento

Integrantes: Fredy Huanca

Jorge Llanos

Edwin Morales

Henry Culi

Profesor: Vicente Machaca Arceda

Fecha de realización: 20 de junio de 2021

Fecha de entrega: 20 de junio de 2021

Arequipa, Perú

Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Índice de Contenidos

1. Preparación de datos	1
2. Implementación de algoritmos en C++ y Python.	2
3. Comparaciones de tiempo de procesamiento.	3
3.1. Comparación de tiempo de procesamiento de algoritmo por lenguaje de programación	3
Referencias	4

Índice de Figuras

1. Comparación de Selection sort en C++ y Python.	3
---	---

Índice de Códigos

1. number_generator.py.	1
---------------------------------	---

1. Preparación de datos

Para la generación de datos en archivos .txt se creó un script de nombre: number_generator.py con el lenguaje de programación Python (Código 1), que genera numeros aleatorios de acuerdo a la cantidad de datos establecida para finalmente crear 9 archivos con nombres que hacen referencia al conjunto de datos que contiene, los archivos generados son:

- random_100.txt
- random_500.txt
- random_1000.txt
- random_2000.txt
- random_3000.txt
- random_10000.txt
- random_20000.txt
- random_30000.txt
- random_100000.txt

Código 1: number_generator.py.

```
1 import os
2
3 RANDBYTES = 1
4 NUM = [100, 500, 1000, 2000, 3000, 10000, 20000, 30000, 100000]
5
6 for i in range(0,len(NUM)):
7     dist = open('random_'+str(NUM[i])+'.txt', 'w+')
8
9     for j in range(0,NUM[i]):
10         ran = int.from_bytes(os.urandom(RANDBYTES), byteorder='little')
11         dist.write(str(ran) + '\n')
12     dist.close()
```

2. Implementación de algoritmos en C++ y Python.

Se han implementado en los lenguajes reueridos los siguiientes algoritmos:

- Bubble sort
- Couting sort
- Heap sort
- Insertion sort
- Merge sort
- Quick sort
- Selection sort

Se pueden apreciar cada uno de ellos visitando el siguiente link: <https://github.com/fredyhuancatorres/G1-MCC-Practica-1>.

3. Comparaciones de tiempo de procesamiento.

Se han realizado las comparaciones de tiempo de procesamiento de cada algoritmo y por cada lenguaje de programación.

3.1. Comparación de tiempo de procesamiento de algoritmo por lenguaje de programación

Se han realizado la comparación de procesamiento del algoritmo Selection sort Figura 1, en los lenguajes de programación C++ y Python, el eje x representa diferentes tamaños de vector a ordenar y el eje y , representa el tiempo de procesamiento.

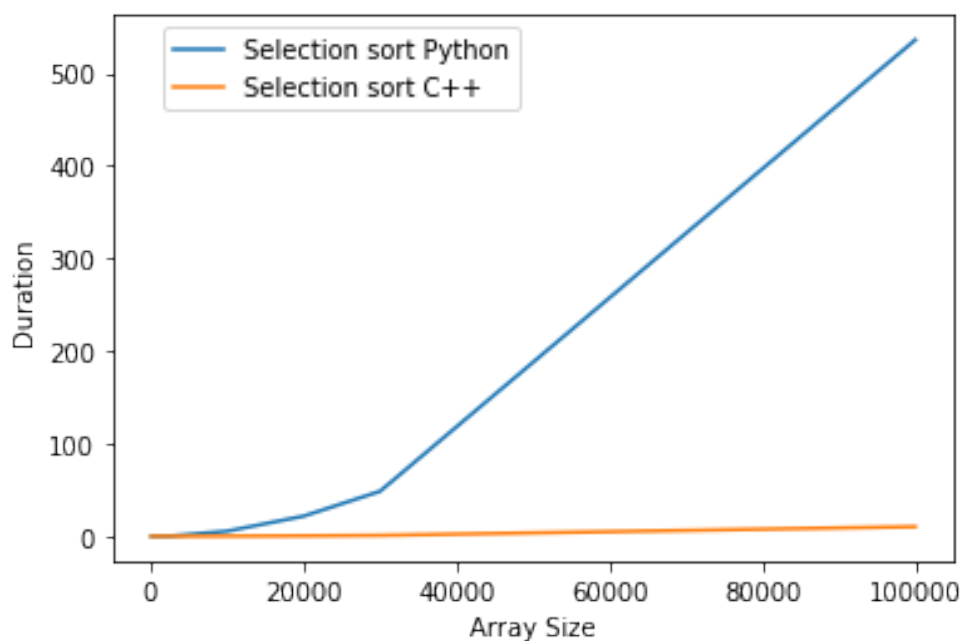


Figura 1: Comparación de Selection sort en C++ y Python.

Referencias

- [1] Template Informe en L^AT_EX. *¡Revisa el manual online de este template!*
<https://latex.ppizarror.com/informe>
- [2] Excel2Latex. *Importa de forma sencilla tus tablas de Excel a L^AT_EX.*
<https://www.ctan.org/tex-archive/support/excel2latex/>
- [3] Overleaf. *Uno de los mejores editores online para L^AT_EX, renovado con su versión 2.0.*
<https://es.overleaf.com/>
- [4] Tables Generator. *Creador de tablas online para L^AT_EX.*
<https://www.tablesgenerator.com>