**School of Computers and Information Engineering**

**STUDENT LEARNING ACTIVITY**

Fall Semester 2022

**Data Structures (SOC 2010)**

Sorting in Metro

Student Learning Assignment

Submitted by

U2110207, Azizbek Muminjonov (Section 002)



**Index Page**

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 1** |

* Problem Introduction
* Operations included in the application
* Choice of Data Structures
* Choice of Algorithm
* Code
* limitations and Future Scope
* Subject Importance
* References

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 2** |

**Problem Introduction**

Nowadays, there are 8 billion people all over the world, as population continues to rise, demand for the faster types of transportation also shows incredible growth. As many people there are in the city, as many cars will be there, which leads to infinity traffic jams. So, people need faster and preferably cheaper type of transportation, and this type of transportation is Metro(Underground). However, sometimes so many people there, which creates even bigger congestions. Usually, cities with their own Metro(Underground) are big enough and people are always busy and in hurry, which leads to mess.

Below is good visualization of this in Tashkent.



According to the official statistics, given in September, Tashkent’s Metro(Underground) carried

out 94 418 300 passengers in 2022 till September, which means about 350 000 people every day, which is incredible.

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 3** |

So, in this kind of crowd even adults come against difficulties when it comes to going in and going out the carriage. Sometimes people even miss their station, due to the size of crowd trying to enter the carriage, even not paying attention to the people inside.

Finally, the only solution to this kind of problem can be gained by implementation of Data Structure Algorithms in order to deal with this kind of huge crowd. There are several methods which allow us to manage and control this situation.

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 4** |

**Operations included in the application**

So, the solution for this situation is Sorting. Sorting people by the amount the stations left till their final destination will definitely make going in and going out easier. For example, passenger A enters the train car and he his destination is the last station. In this situation, people in real life do nothing special, they just stand at the place which is empty and when it comes to other passenger B go out of the train, B meets the obstacle and wastes his and others’ time.



So, what about after sorting, scenario is - A enters the carriage, stands to the empty place and as soon as doors are closed, the person behind him, B, starts to ask about his destination and if they find out that B goes out before the A, they swap their places. It is actually working, I always do the such way before exiting the carriage and still regardless who was in front of me, no one refused me to swap the places. So, it seems if to give some simple and natural algorithms and start to teach to it people it is easy to get preferred result at the end.

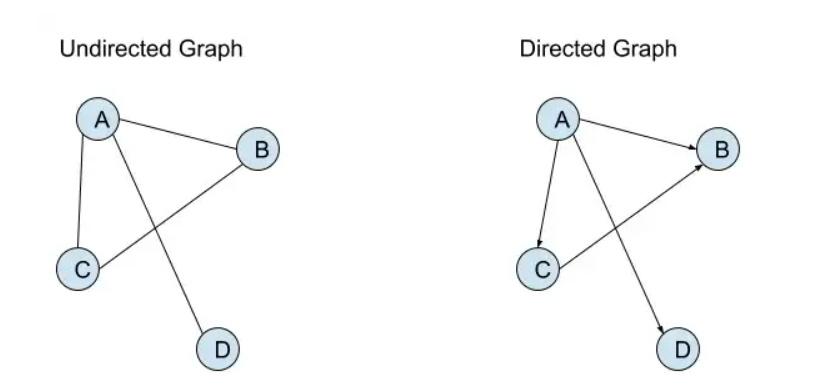
|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 5** |

**Choice of Data Structure**

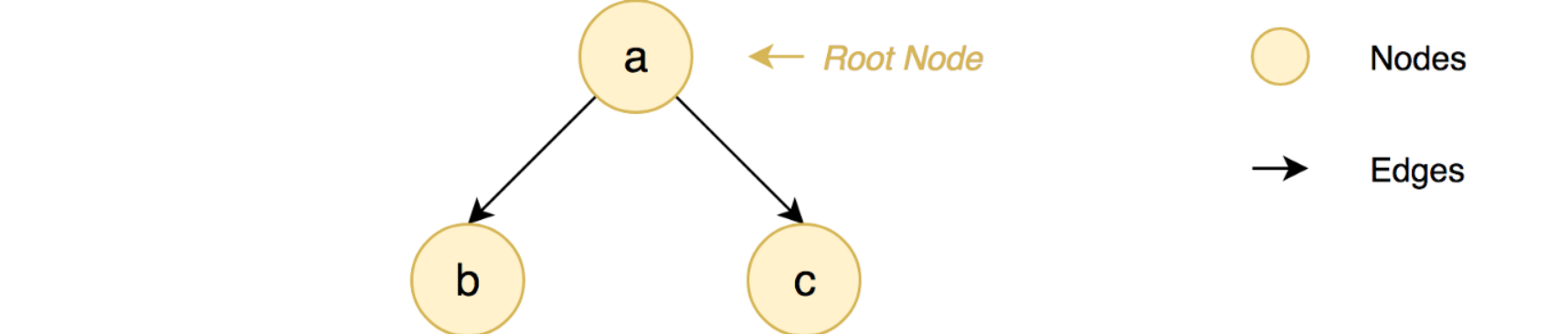
Turning into the details of solution, mentioned above, among all Data Structure Algorithms the

most relevant to our problem – are Sorting Algorithms, which are used by any Data Structure, Arrays, Binary Trees, Graphs and etc. However, my solution is based on Sorting Algorithms implemented to Arrays. There are several reasons for my choice.

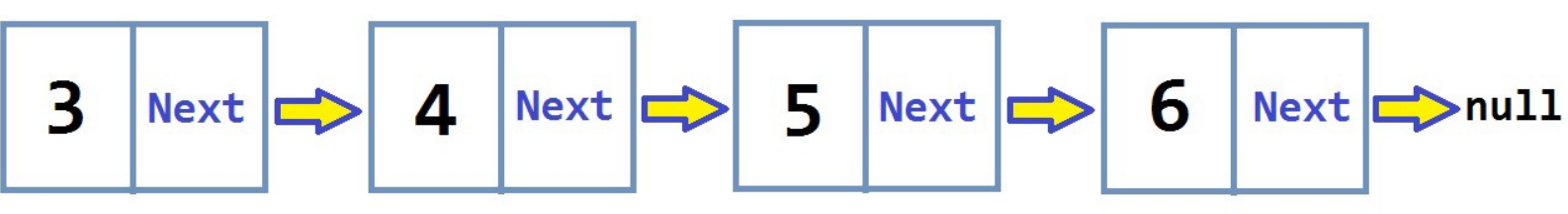
First of all, as we deal with train cars implementation of Graphs, except for train’s path, is simply unlogic, because Graphs are usually used for GPS navigation, as they have direction, used for Social Media networks, Recommendations and etc., but not for the sorting or managing big data, in our case people.



Secondly, Binary trees are also very good with sorting, dealing, accessing, inserting and deleting the data, even better than linked list or arrays, however it is designed more for storing hierarchical data and it has no limitations on space and grow till infinity, and when it comes to train cars, where place is restricted, it may provoke troubles.

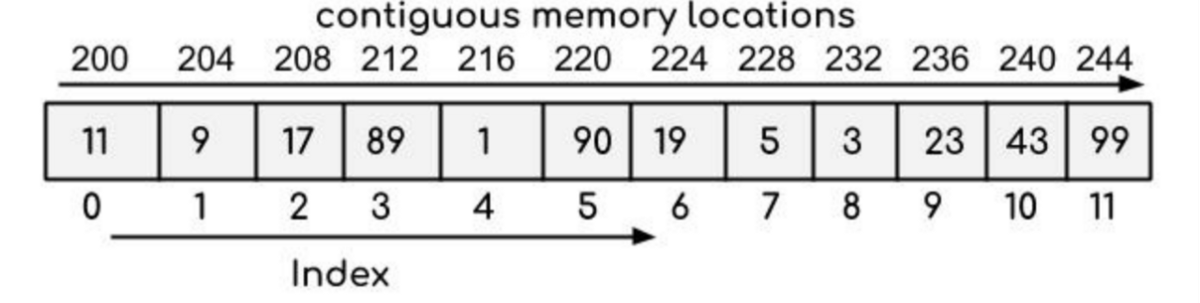


Finally, Linked lists are also one type of Data Structure, which can grow infinitely. And via Linked Lists accessing the data is a bit time consuming rather than Arrays and Binary Trees.



|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 6** |

Overall, Arrays are the most appropriate Data Structure. Firstly, arrays are static data structure and we must know in advance how much space should be allocated, in the normal situation it would be a disadvantage of this data structure, but not in our case, as we have total capacitance of train cars and we physically can not fill the train the over quantity of passengers. And also, usually Arrays contain only one type of elements, like string, integer, float or etc. , so in our case we input only people, too. Concept of Arrays are very primitive and any person intuitively get the point on the move. That’s why Arrays are the best choice.

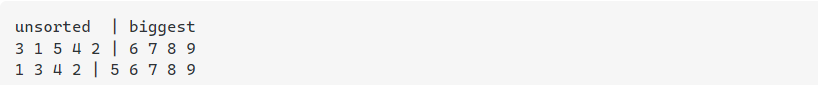
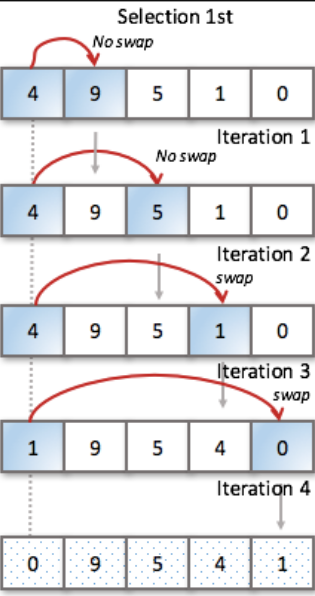


|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 7** |

**Choice of Algorithm**

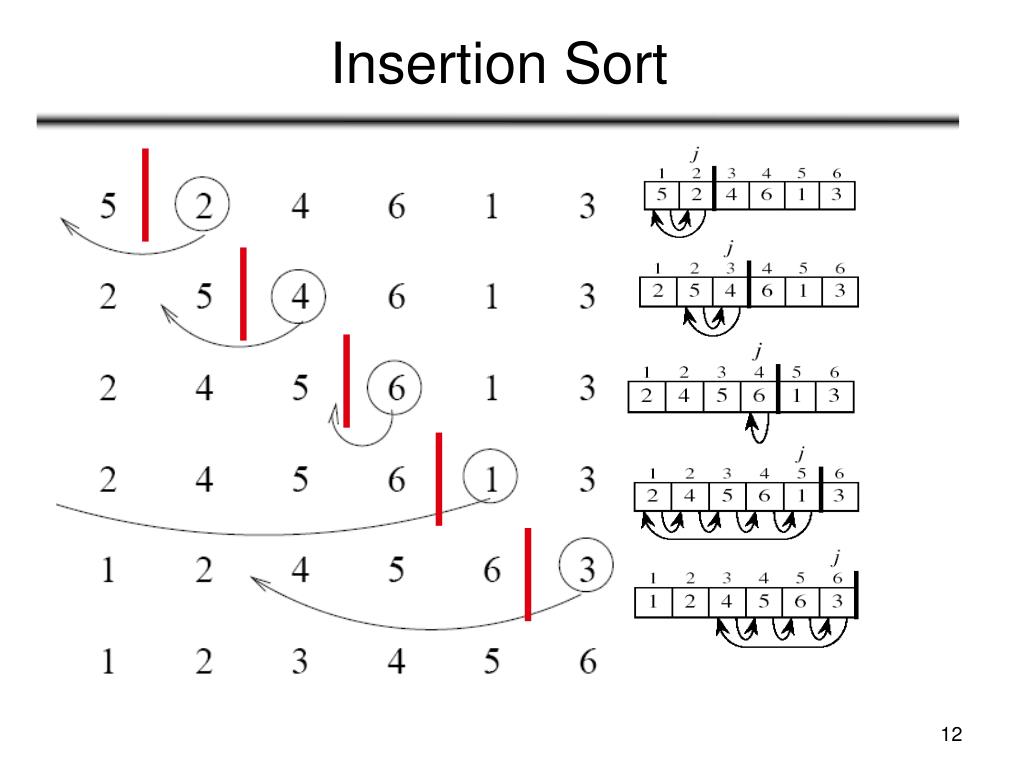
The Algorithm we need to implement is sorting algorithm. There are several sorting algorithms in the Data structure and all of them are good at something and bad in another. The research made by me showed that only one algorithm can be easily implemented in real life, very close to the scenario I have provided above and people actually do this thing usually even have not recognized what it is.

Bubble sort, first of all I thought that it is the ideal data structure algorithm to implement as a solution, but then I have found out that the way how it works is a bit far from something would basic person have done subconsciously everyday. Because it goes repeatedly through out the list and is time consuming.

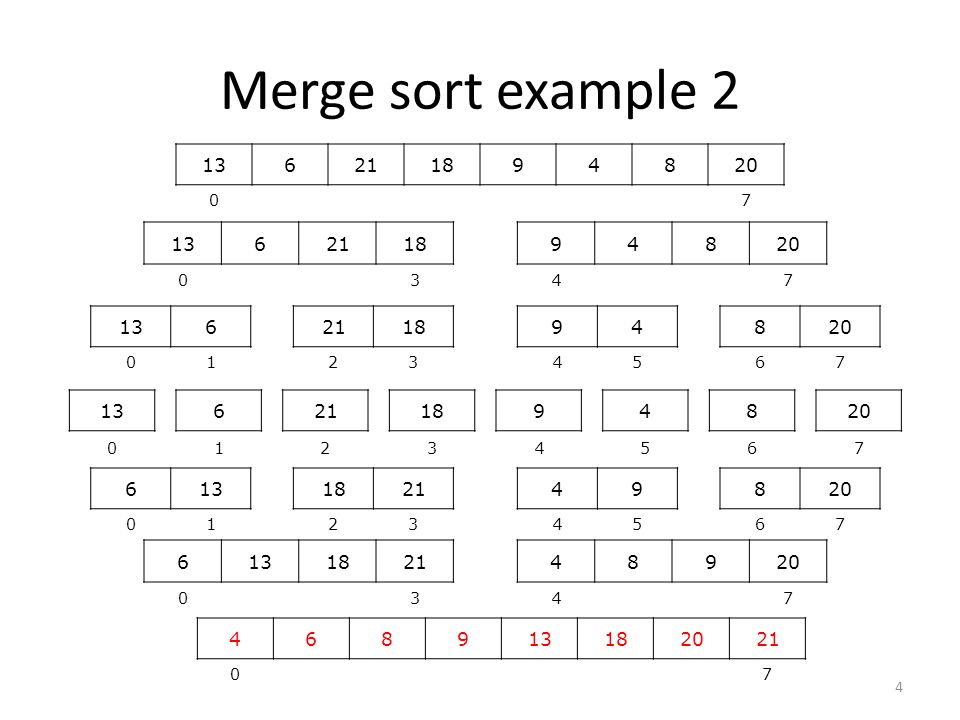


Selection sort, this method requires scanning and in order to do it we have to hire conductor to check passengers locations, which is even not a solution. So, this algorithm is also not for us.

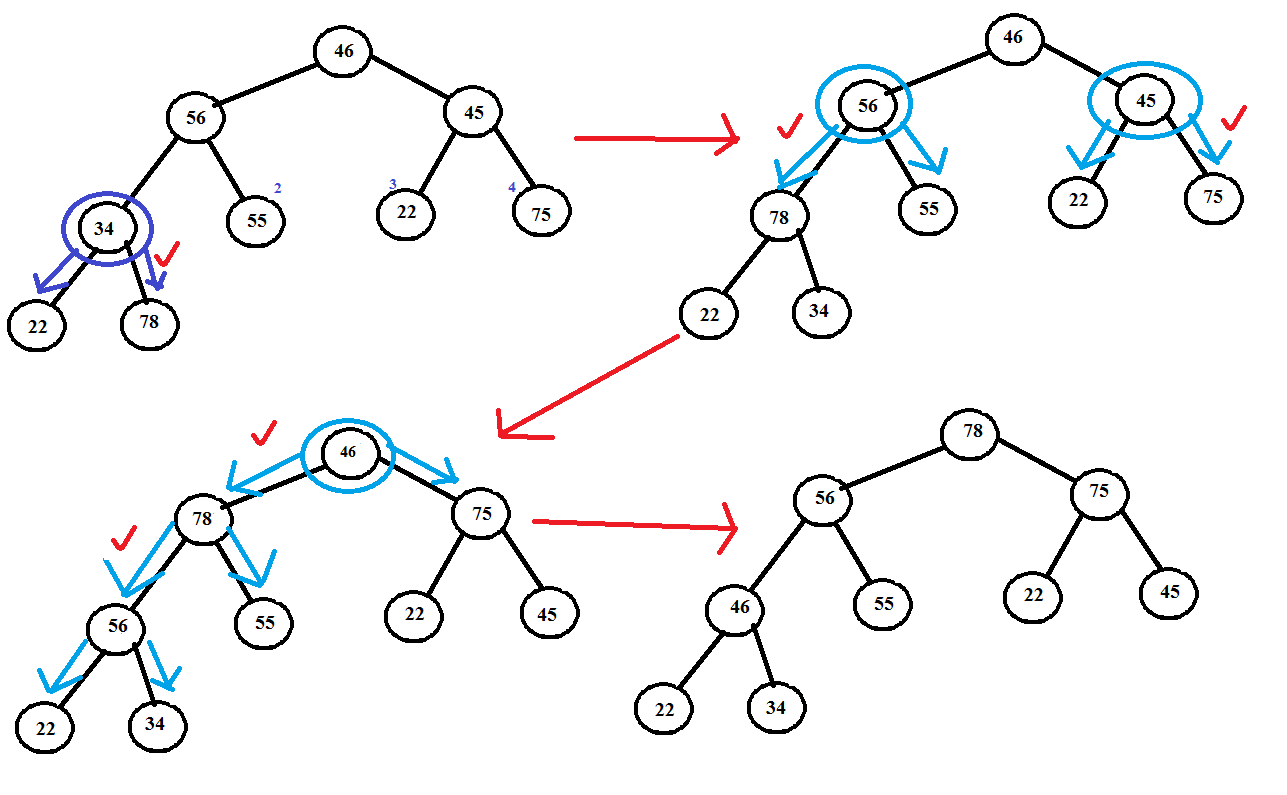
Finally, Insertion sort, this method comes about the best solution among others. The way, how it works looks like Bubble sort, but this method is twice as fast as Bubble sort, and while sorting item is taken and the same item goes till the end while it is the biggest one, just like a passenger just asks and swaps and so on.



|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 8** |

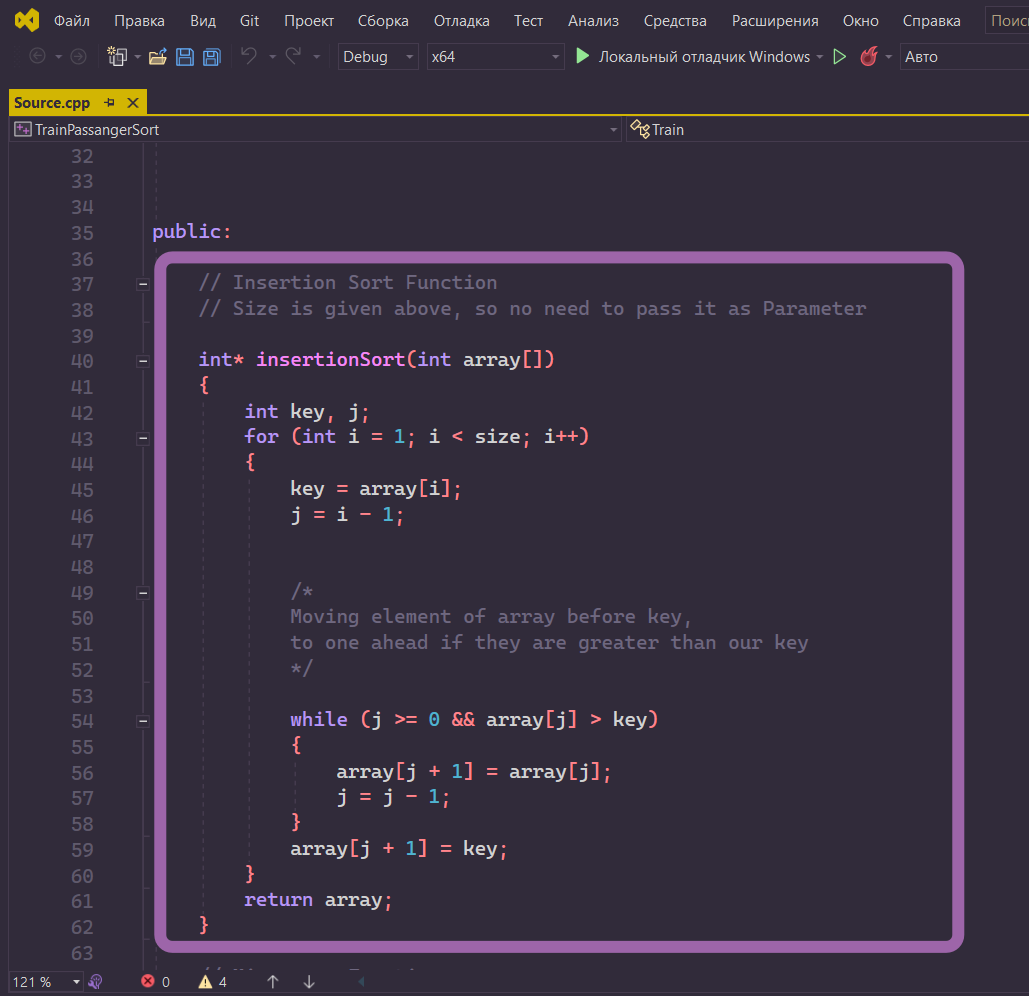
Overall, what about others, like Merge sort, Heap sort, quick sort, they all are not this case, research has shown that’s ones are difficult to implement in real life or very difficult for normal people, others require some additional things like binary tree or something like this. 

Heap sort



|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 9** |

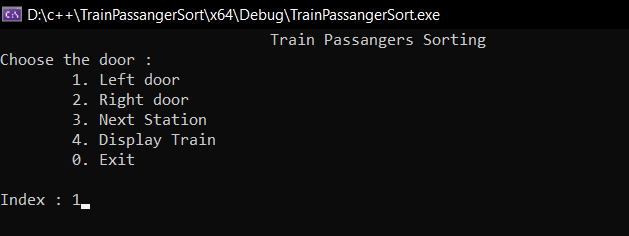
**Code**

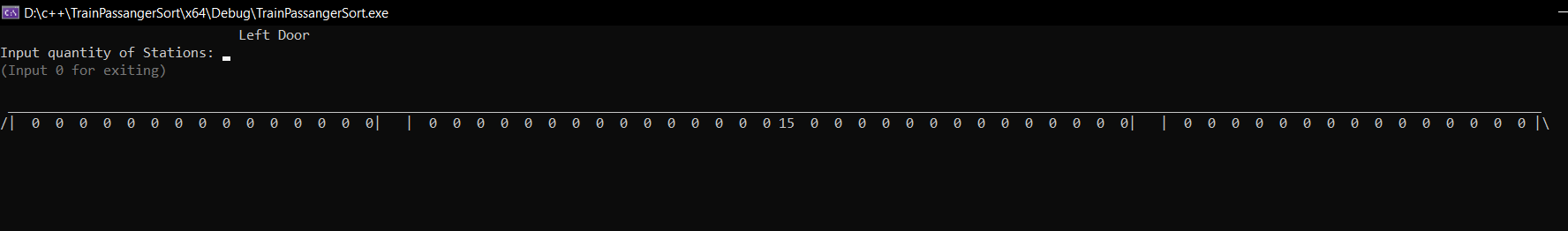
The code, program is written in C++ and recommended to run in Visual Studio 2019+ versions.

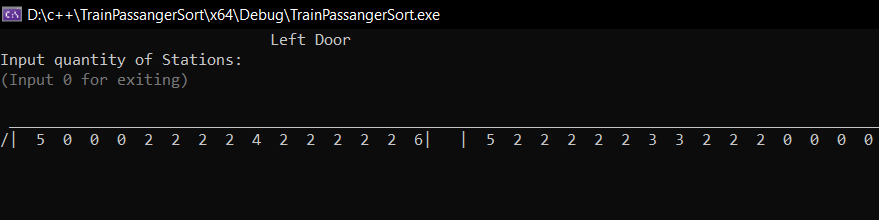
Above, the Insertion Sort function, which takes an array as a parameter and returns the sorted array, the way how it works explained on comments.

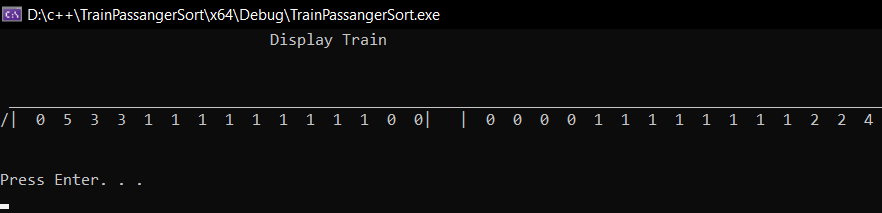
The full Program is available on my Git Hub, link to which I left on Reference page and additionally provided QR-Code to easier access.

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 10** |

 A bit about the program, start, input and output.

Input of 1 opens Insert Window

Now Train is empty as Station is 1, lets insert some numbers, I will cut half of train in the screen to make output more visible

Then train goes to the next station, on the way Insertion Sort function is called and From all subtracts 1 on every Station

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 11** |

**Limitations and Future scope**

The only Limitation for implement this algorithm is people their own. Because everything is very simple and there is no need to be super clever to understand it. Even if the person, in our case passenger is kind of introvert, who does not want to interact with other people, if the other people will anyway follow to this method, our passenger is always able to swap the place regardless of his personality.

About the Future scope, if to start to implement this algorithm on a daily basis, and teach our children and siblings to this solution, it will definitely become a habit, like greeting people before talk to them.

So, lets change the world together.

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 12** |

**Subject importance**

I think, without Data structure it would be very difficult to come to certain general solution, to this kind of problems, which require constant dealing with a big data. Data structures and its’ algorithms help us visualize the issue, to simulate them with programming and finally find out some kind of set of solutions, which can be implemented anywhere depending on the demand and requirements.

Overall, this subject can definitely change world in the right hands and right goals.

|  |  |
| --- | --- |
| **Sorting in Metro** | **Page No 13** |

**References**

Official Statistics:

<https://kun.uz/ru/25201130>

https://www.gazeta.uz/ru/2022/12/06/metro/

Array:

<https://medium.com/geekculture/the-array-data-structure-explained-b8eb4c5d1f7a>

Graphs:

https://medium.com/swlh/data-structures-intro-graphs-af165b1e2c95

<https://www.upgrad.com/blog/graphs-in-data-structure/>

Binary Trees:

https://www.geeksforgeeks.org/binary-tree-data-structure/

Linked Lists:

<https://medium.com/basecs/whats-a-linked-list-anyway-part-1-d8b7e6508b9d>

https://www.youtube.com/watch?v=F8AbOfQwl1c

Insertion sort:

<https://www.youtube.com/watch?v=JU767SDMDvA>

Selection sort:

<https://www.youtube.com/watch?v=g-PGLbMth_g&t=52s>

Bubble sort:

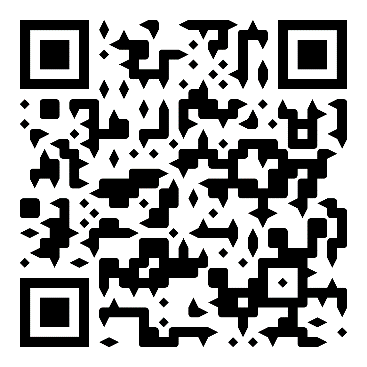
<https://www.youtube.com/watch?v=xli_FI7CuzA&t=31s>

Merge sort:

<https://www.youtube.com/watch?v=4VqmGXwpLqc>

Heap sort:

<https://www.youtube.com/watch?v=2DmK_H7IdTo&t=84s>

Quick sort:

<https://www.youtube.com/watch?v=Hoixgm4-P4M&t=200s>

Code link to git hub:

<https://github.com/Black-Spades-Z/Data-Structure.git>