

Problems: UNIT 2

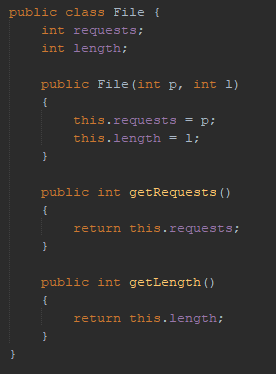
VORACIOUS ALGORITHMS



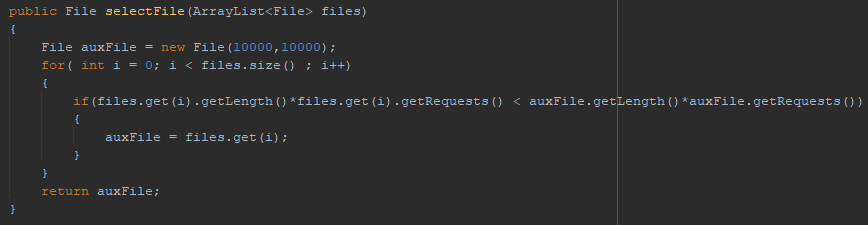
JAVIER GARCÍA JIMÉNEZ AND ISABEL MARTÍNEZ GÓMEZ

EXERCISE 2

We have created a class File that stores the number of the requests corresponding to the file and the length of the file.

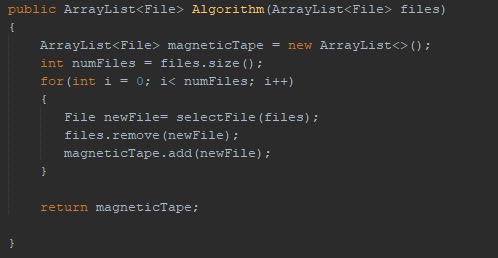


The function selectFile returns the file whose quantity is the lowest one from the set of files on the magnetic tape. This is the algorithm that selects the best candidate of voracious algorithm.

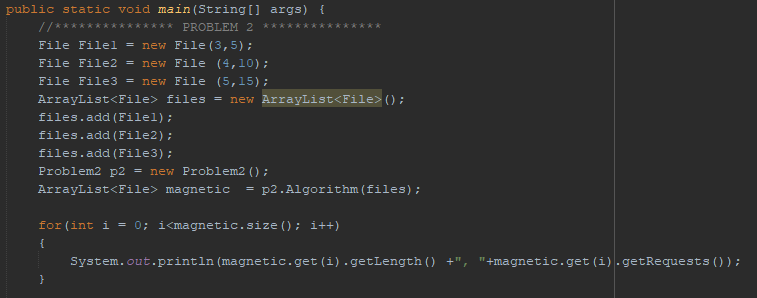


Finally, we have our voracious algorithm to solve this problem. In this function there is an entry of an array of files with size n that are going to be part of the solution.

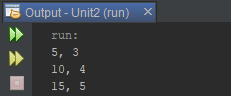
To get the optimal solution, which is going to minimizes the quantity, in each step we try to add a file to the *ArrayList magneticTape* using the previous function explained. The algorithm ends when we have all the files in the array so that the optimal solution is reached.



In the main class we have created some Files with its requests and length.

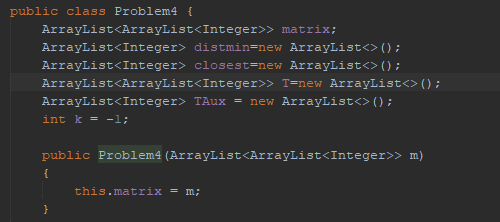


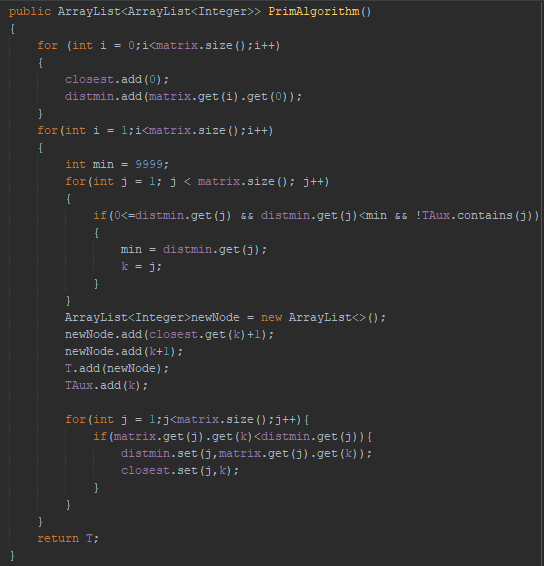
The result of this is the following:



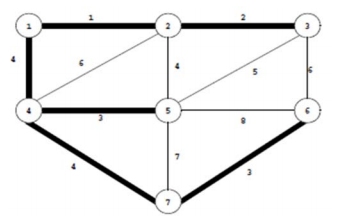
As we can see this solution is the optimal one because it minimizes the average loading time.

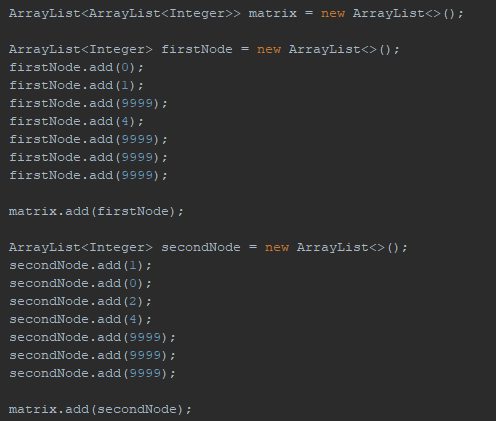
EXERCISE 4

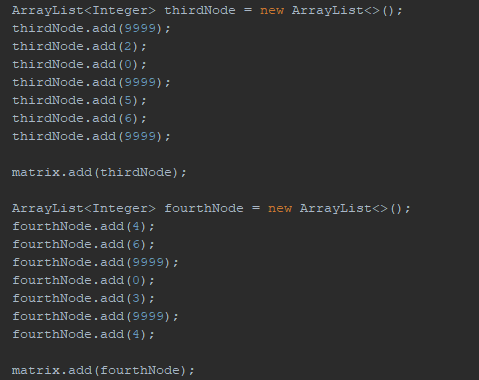


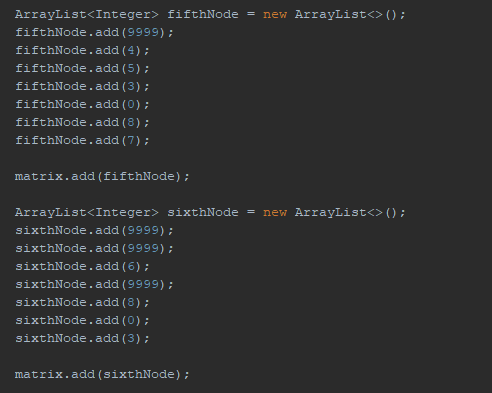


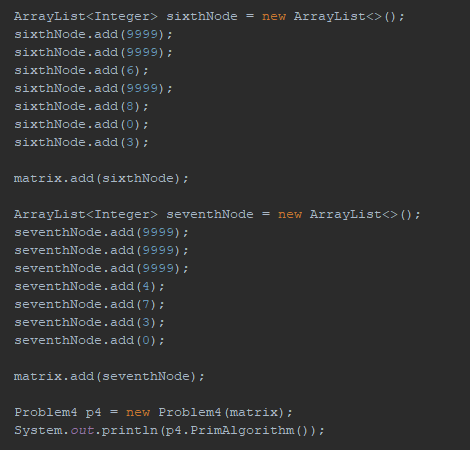
This is the main function where we are creating the following graph.



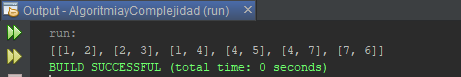








Finally, this is the result of the example executed in the main function.



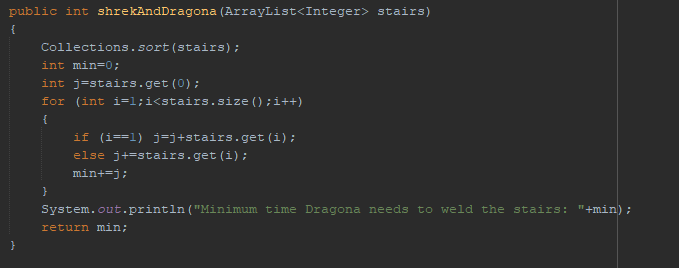
EXERCISE 6

In the attached image we have the voracious algorithm implemented where we calculate the minimum cost to weld all the stairs in the vector.

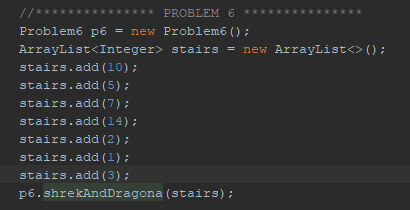
To do this algorithm, the first thing we will have to do is order the stairs arraylist so that we can obtain the minimum time possible.

Then, we go through the arraylist stairs adding each first, the first stair with the second, then it is added the result of the first and the second with the third stair and so on until all the stairs are added.

Finally, we return min that is the minimum cost to weld all the stairs in the arraylist.



To prove it, we have considered that the arraylist **stairs** has the following length of stairs:



We get the result:

