

## Report Assignment 2 Data Structures

---

For this assignment we have been practicing sorting and the different ways to implement it in our program. This is what we have done in the first two tasks.

On task 3 we have been asked to add some variables to our sorting functions to see the amount of comparisons and swaps we do in each function. And below here we can see the results we have gotten:

Arrays of size 10000:				
Selection sort				
sorted	TEST	SORTED	SWAPS	COMPS
sorted	Ascending	YES	0	49995000
sorted	Descending	YES	25005000	50000000
sorted	Uniform	YES	0	49995000
sorted	Random w duplicates	YES	85897	50004993
sorted	Random w/o duplicates	YES	87617	50004985
Insertion sort				
sorted	TEST	SORTED	SWAPS	COMPS
sorted	Ascending	YES	0	9999
sorted	Descending	YES	49995000	50004999
sorted	Uniform	YES	0	9999
sorted	Random w duplicates	YES	25093231	25103230
sorted	Random w/o duplicates	YES	24863510	24873509
Quick sort				
sorted	TEST	SORTED	SWAPS	COMPS
sorted	Ascending	YES	50004999	49995000
sorted	Descending	YES	25004999	49995000
sorted	Uniform	YES	9999	49995000
sorted	Random w duplicates	YES	78162	155320
sorted	Random w/o duplicates	YES	76608	149413

As we can see, depending on the function and the type of array we have, some sorting methods are better than other. For example, if we are implementing an array with random numbers with or without duplicates. The best sorting method is the quick sort. But if we are implementing an ascending or uniform array. The best method is the insertion sorting. This shows us how we should evaluate our array type before choosing what method we are going to apply to sort our array. This way, if we select the most efficient we will save memory and time.

Finally, for task four, we are asked to implement a program in which we evaluate the games of the last 20 years and rank them. The method I have used to approach this program is pretty simple. Explaining it in a simple way, I have created a structure in which I store the information from each game, like name, platform... Then I have a compare function I am using to compare the games and rank them. This all gets done using the qsort function, which using our structure and compare function will rank our games regarding their score. Lastly, to complete the program and make it cleaner. I have added a usedTitle function that will also check if the title is already in our top ranking. This will allow us to not see the same game but in different platforms in our ranking. The reasoning behind this, is that in the resulting Top 10, I had four versions of a Metal Gear Solid game, all receiving a 10 as it is the same game. So I created this function so that wouldn't happen.

The last question we are asked to theoretically approach in this assignment is how we would get the Top 10 games for each of the last 20 years. There are many ways we can approach this issue. But one obvious but maybe tedious approach that we could use is when printing our games. We could selectively print the Top 10 for each year. This could be a bit tedious to write as we would be repeating the same line again and again 20 times, but it would work.