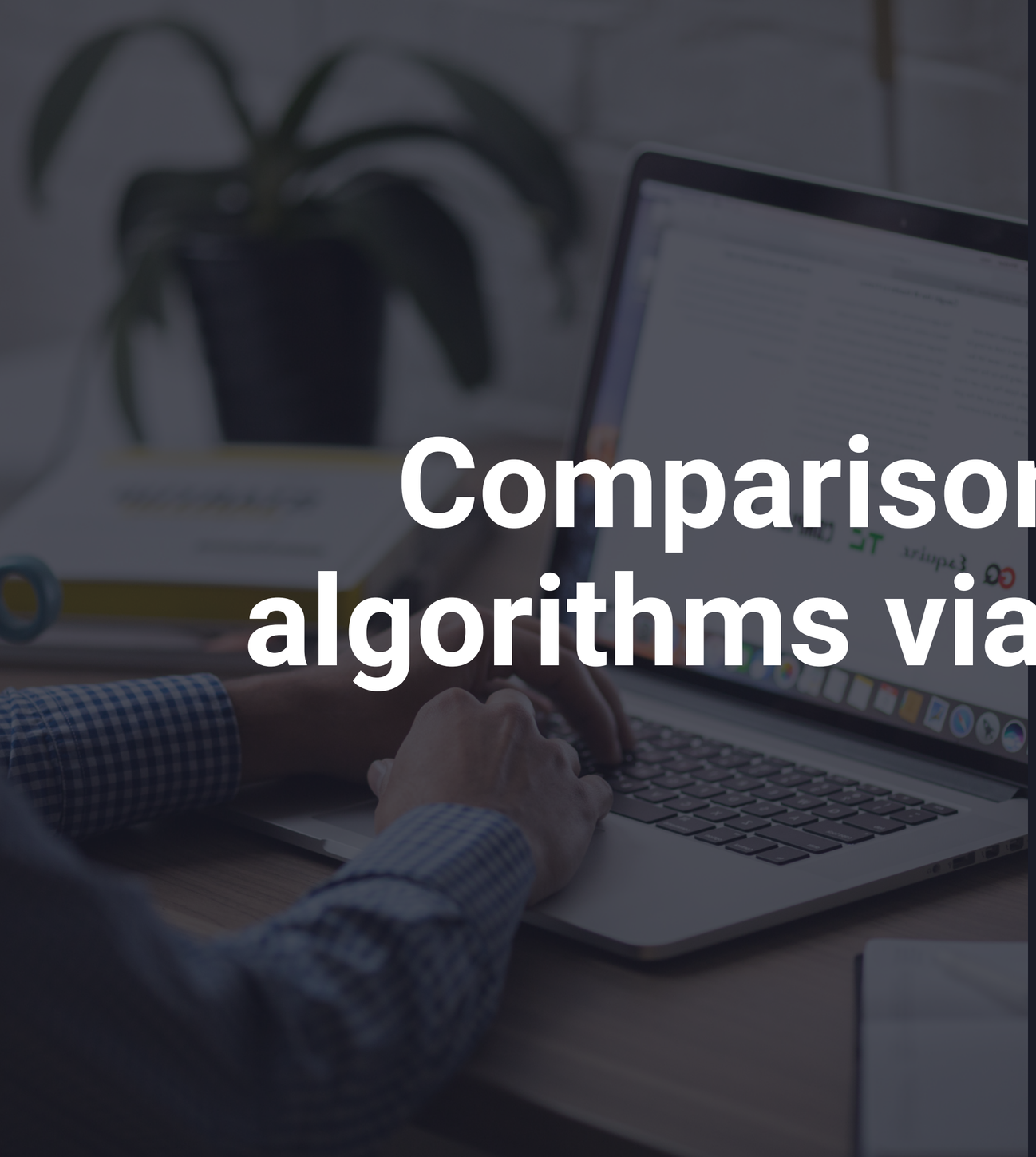


Comparison of sorting algorithms via Visualization

Boranbayev Daulet 21MD0234
Turdaleyev Azamat Farabi 21MD0241

We took the most popular sorting algorithms:

- BUBBLE SORT
- COCKTAIL SORT
- SELECTION SORT
- INSERTION SORT
- MERGE SORT
- QUICK SORT
- TREE SORT

BUBBLE SORT

The idea of sorting is very simple, one by one the neighboring elements of the list are compared, if the first compared element is larger than the second, we swap their places. Thus, the highest items are moved to the end (pop up).

COCKTAIL SORT

Shuffle sorting is similar to bubble sorting, during bubble sorting we move only one way, while shuffle sorting moves both ways, so the largest items move to the end and the smallest items move to the beginning.

SELECTION SORT

In the list the smallest element is searched, then its position changes with the first unsorted position, in the next iteration the search does not include the sorted list. In this way the whole list is sorted.

INSERTION SORT

The elements of the input sequence are reviewed one by one, and each new incoming element is placed in a suitable place among the previously ordered elements

MERGE SORT

Recursively sorts the array halves and then combines them into one

QUICK SORT

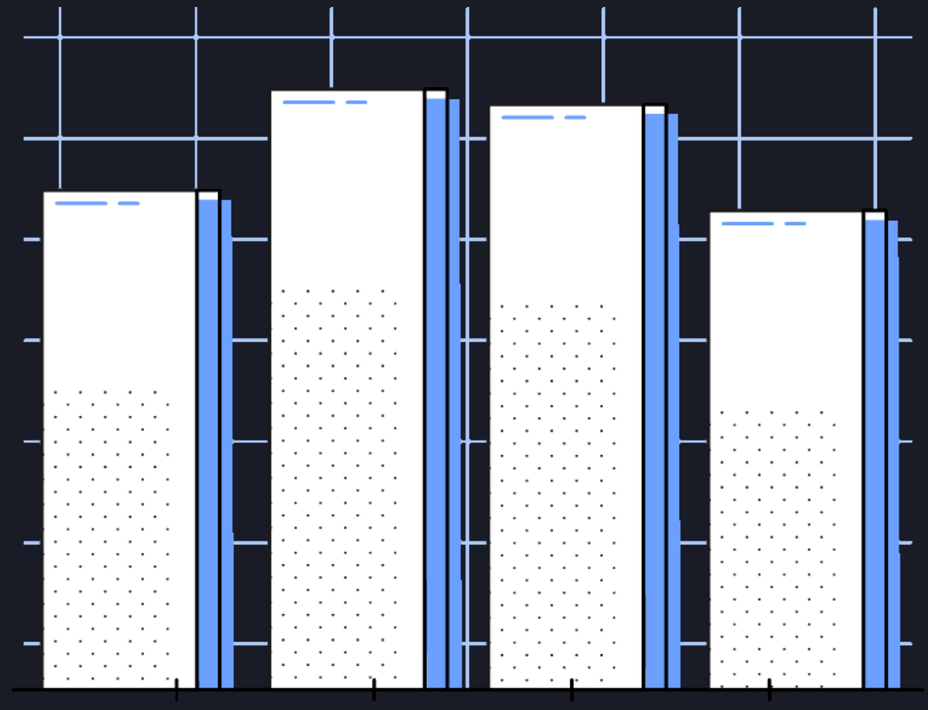
A reference element p is chosen. All keys smaller than p are moved to the left of it, and all keys larger or equal to p are moved to the right. Then the algorithm is recursively applied to each of the parts

TREE SORT

Based on the initial data, a binary search tree is built, which sequentially collects the minimum values

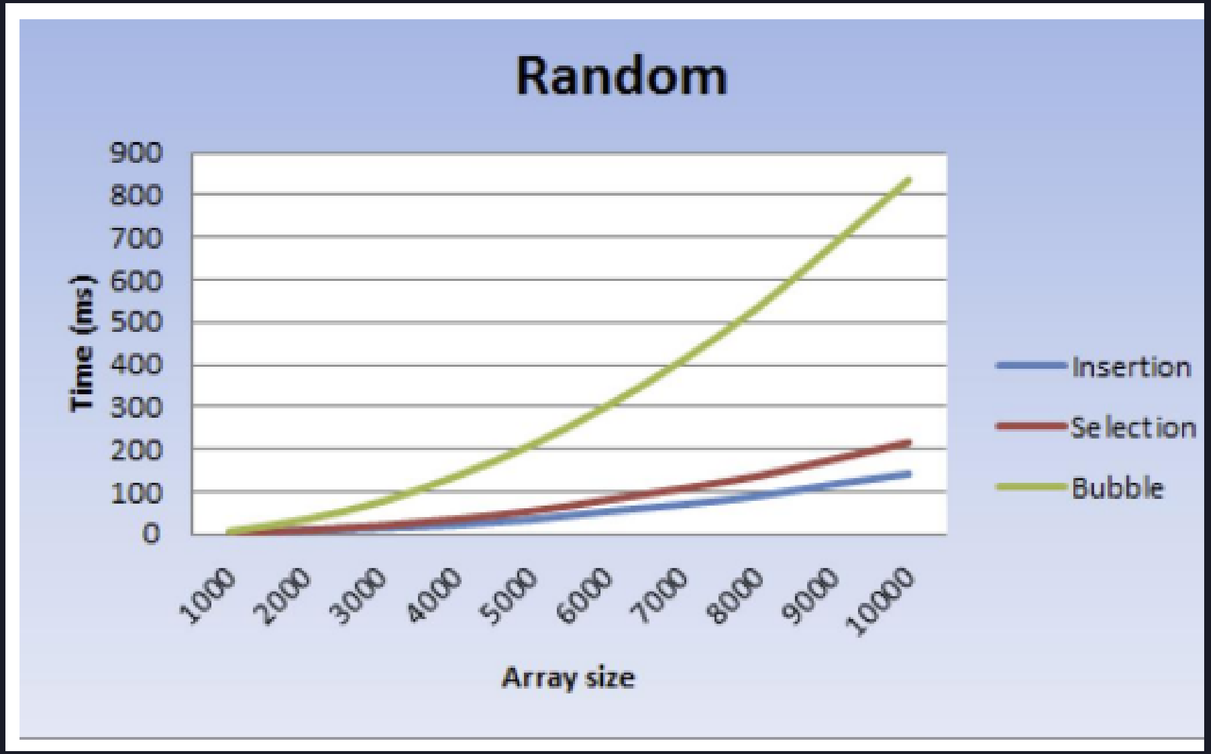
Tasks

- Principle of operation
- Comparison analysis
- Visualization



BACKGROUND

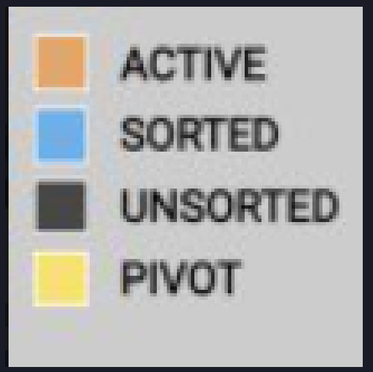
FIGURE 6 DEMONSTRATES THAT THE RUNNING TIME OF THE ALGORITHM GROWS FASTER FOR BUBBLE SORTING AND SLOWER FOR INSET SORTING. IF WE CONSIDER COMPARING RUNNING TIMES FOR AN ORDERED ARRAY, THE PREFERENCE SHOULD ALSO BE GIVEN TO THE BUBBLE SORTING ALGORITHM. BUT IF YOU NEED TO SORT AN ARRAY WHERE THE ELEMENTS ARE IN REVERSE ORDER TO BE SORTED, SORTING BY CHOICE OR INSERTS IS MORE EFFICIENT THAN BUBBLE SORTING.

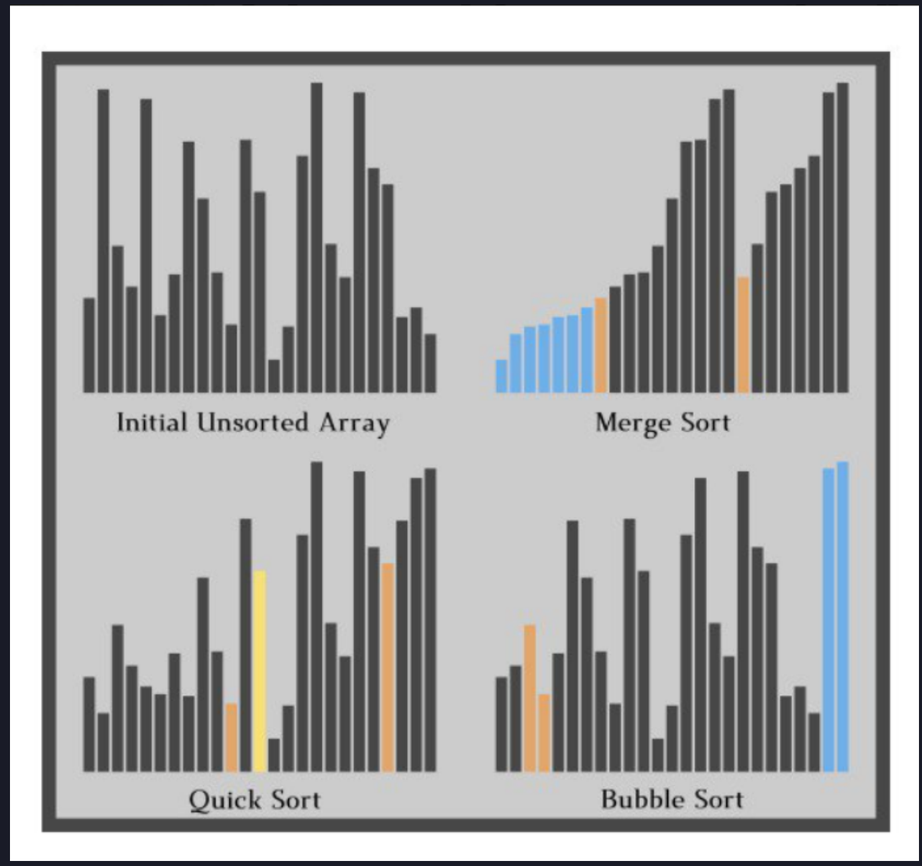


SORTING ALGORITHMS. ANALYSIS, REALIZATION, APPLICATION.
D.V. SHAGBAZYAN, A.A. SHTANYUK, E.V. MALKINA,
2019

BACKGROUND

Here is visual differences between various sorting algorithms with the same set of data. The difference in visualization shows the approach each algorithm takes to sort the data. The merge sort and quick sort algorithms are based on the divide and conquer strategy while the bubble sort is based on the iterate and compare strategy.





Bikram Karki

Sorting Algorithm Visualization 08 November 2021

BACKGROUND

Figure 4. Sorting comparison for worst case.

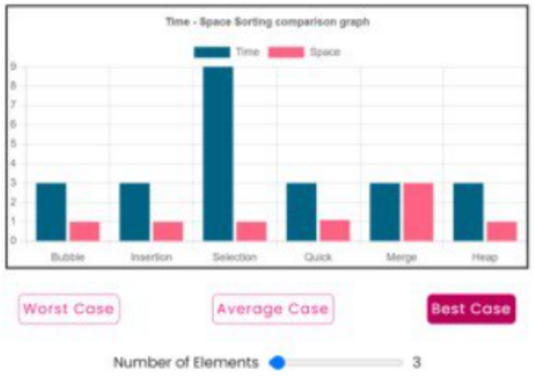


Figure 5. Sorting comparison for best case.

THERE ARE BASICALLY THREE CASES FOR EVALUATION OF THE EFFICIENCY OF AN ALGORITHM, I.E., THE AVERAGE CASE, WORST CASE AND BEST CASE. WE INCLUDED THESE SCENARIOS FOR THE ALGORITHMS TO WORK UPON. A WORST-CASE OCCURS FOR AN ALGORITHM WHEN IT TAKES MAXIMUM TIME TO COMPLETE IN A SITUATION OR ON A SET OF DATA. FOR EXAMPLE, THE PERFORMANCE OF BUBBLE SORT WILL BE WORST WITH TIME COMPLEXITY O LOG N WHEN THE INPUT ARRAY IS REVERSELY SORTED AS CLEARLY VISIBLE IN THE FIGURE 4. A BEST-CASE SCENARIO AS SHOWN IN THE FIGURE 5 IS WHEN AN ALGORITHM PERFORMS ITS TASK UNDER SMALLEST AMOUNT OF TIME IN SOME CONDITIONS. JUST LIKE THE PERFORMANCE OF BUBBLE SORT WILL BE BEST WHEN THE INPUT ARRAY IS ALREADY SORTED. THE AVERAGE CASE IS THE CLOSEST TO THE EXPECTED RESULT IN THE REAL WORLD. IT IS CALCULATED BY TAKING ALL THE POSSIBLE INPUTS AND TAKING THE AVERAGE OF THE TIME AND SPACE TAKEN. LIKE THE UNSORTED ARRAY IS GIVEN INPUT TO THE BUBBLE SORT TO IMPLEMENT.

ALGORITHM VISUALIZATION - MODERN WEB-BASED VISUALIZATION OF SORTING AND SEARCHING ALGORITHMS
AUTHORS: ASHISH KUMARI, MANAV MITTAL, VIPUL JHA, ABHISHEK SAHU, MANISH KUMAR, NEETI SANGWAN AND NAVDEEP BOHRA,
2021