|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SORTING ALGORITHMS** | | | | | |
|  | **Tournament** | **Quick** | **Merge** | **Strand** | **Heap** |
| **Who** | Alexander Stepanov | Tony Hoare | John von Neumann | John Cohen (mention) | John Williams |
| **When** | 2002 | 1959 / 1960 | 1945 | 1997 | 1964 |
| **What** | * A variation of heapsort which improves upon the naïve selection sort by using a priority queue to find the next element in the sort. | * a popular sorting algorithm that is often faster in practice compared to other sorting algorithms. Quickly sort data items by dividing a large array into two smaller arrays. | * Sorts by breaking it into halves, sort, and then merge together | * It repeatedly pulls out a series of elements from the unsorted list into a sub list to be sorted, then merges them into the result array. * It is derived from Selection Sort. | * It is similar to the selection sort where we first find the minimum element and place the minimum element at the beginning. |
| **Used** | Uses selection | Uses partitioning. Used when stable sort is not needed to search for something. | Uses merging  Can be done in-place but increases time complexity. Best on large data. | Generally used to sort elements in ascending order. | Uses selection |
| **Conditions** | * Unstable algorithm * Out-of-place sorting * Comparison approach | * Unstable algorithm * In-place sorting * Divide and conquer * Comparison approach | * Stable algorithm * Out-of-place sorting * Divide and conquer algorithm * Comparison approach * recursive | * Stable algorithm * Out-of-place sorting * Comparison approach | * Unstable algorithm * In-place sorting * Comparison approach |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TIME COMPLEXITY** | | | | | |
|  | **Tournament** | **Quick** | **Merge** | **Strand** | **Heap** |
| **Best** | O (n log n) | O (n log n)  Pivot = middle element | O (n log n) | O (n) | O (n log n) |
| **Average** | O (n log n) | O (n log n)  Pivot = random | O (n log n) | O (n log n) | O (n log n) |
| **Worst** | O (n log n) | O (n2)  Pivot = least/greatest | O (n log n) | O (n2) | O (n log n) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPACE COMPLEXITY** | | | | | |
|  | **Tournament** | **Quick** | **Merge** | **Strand** | **Heap** |
| **Worst** | O (n) | O (1) unstable partition  O (log n) tail recursion | O (n) | O (n) | O (1) |