
DSA Lab Assignment-9 (March 3rd week 2019)

Write a report on the following topics and submit a soft copy in institute moodle (web-link will be posted later). Do not require to submit your programs for this assignment. Also, write your name and roll number in the report.

1. Provide the average running times of the following 6 algorithms on 10^5 elements in tabular format with 6 rows and single column. Comment on the threshold value in optimization technique-1.
 - (a) Implement merge-sort algorithm.
 - (b) Implement merge-sort algorithm with optimization technique-1.
 - (c) Implement merge-sort algorithm with optimization techniques 1 and 2.
 - (d) Implement merge-sort algorithm with optimization techniques 1,2 and 3.
 - (e) Implement merge-sort algorithm with 4 optimization techniques.
 - (f) Implement merge-sort using bottom-up approach, with out using recursion.

Optimization Techniques:

1. Terminate recursion at an early stage when the size of the array is below a **threshold** and sort them using insertion sort. Experiment multiple ways to find a suitable value for threshold (powers of 2).
 2. When we merge the elements in A and elements in B , if the last element in A is less than the first element in B , we can ignore the merging.
 3. Eliminate the creation of arrays locally inside the merging.
 4. Pass two arrays A and B in merge-sort, and process the elements in A and keep the resultant sequence in B , if recursion level is an odd number; otherwise process the elements in B and keep the resultant elements in A .
2. Provide the average running times of the following 2 algorithms on 10^5 elements in tabular format with 2 rows and single column.
 - (a) Implement quick-sort algorithm by considering leftmost element as a pivot.
 - (b) Implement quick-sort algorithm by considering the median of three elements $A[1]$, $A[n/2]$ and $A[n]$ as a pivot, where $A[i]$ denotes the i th element in the input array A on n elements.
 3. Provide the average running times of the following 6 algorithms on 10^3 , 10^4 , 10^5 and possibly on 10^6 elements in tabular format with 6 rows and 4 columns.
 - (a) Implement insertion-sort algorithm.

- (b) Implement selection-sort algorithm.
- (c) Implement bubble-sort algorithm.
- (d) Implement the merge-sort algorithm using all the 4 optimization techniques.
- (e) Implement the quick-sort algorithm using median-technique.
- (f) Run the `sort` function using standard template library.

Note: Average running time can be obtained by running the same algorithm on 10 different test cases that contain same number of elements. You can create a test case on n numbers as follows: Consider the first n natural numbers and rearrange them randomly, using `random_shuffle` function in standard template library.