

GAPO DESIGN and ANALYSIS OF ALGORITHMS — Assignment 1 Report

GAPO DESIGN and ANALYSIS OF ALGORITHMS — Assignment 1 Report

1) Architecture Notes

- MergeSort: reusable buffer to avoid per-merge allocations; small- n cutoff to insertion sort; skip-merge optimization when $\text{middle} \leq \text{middle}+1$.
- QuickSort: randomized pivot; “smaller-first recursion, larger-side iteration” to keep recursion depth $\approx O(\log n)$ in practice.
- Select: Median-of-Medians (groups of 5); compact group medians; pivot is median of medians; recurse only on the side containing k .
- Closest Pair: pre-sort by x and y ; recursive split; strip built in y -order; scan next up to 7 neighbors per point.

2) Recurrence Analysis (sketch)

- MergeSort: $T(n)=2T(n/2)+\Theta(n) \Rightarrow \Theta(n \log n)$ (Master Case 2).
- QuickSort (average): $T(n)=T(\alpha n)+T((1-\alpha)n)+\Theta(n)$; randomized pivot \Rightarrow expected $\Theta(n \log n)$; depth $\Theta(\log n)$ with smaller-first recursion.
- Select (MoM-5): $T(n) \leq T(n/5) + T(7n/10) + \Theta(n) \Rightarrow \Theta(n)$.
- Closest Pair: $T(n)=2T(n/2)+\Theta(n)$ for split+strip $\Rightarrow \Theta(n \log n)$.

3) Metrics & Plots

CLI records CSV rows: algo,n,time_ns,depth,comparisons,moves.

Generate datasets via:

```
mvn -q exec:java -Dexec.args="--algo mergesort --n 100000 --trials 3 --csv results.csv"
```

Repeat for quicksort, select, closest with various n .

Plot time vs n and depth vs n from results.csv.

4) Summary

Asymptotic results match expectations: Merge/Closest $\sim n \log n$ growth; Select \sim linear trend; QuickSort $\sim n \log n$ average with modest depth due to smaller-first recursion. Constant factors vary with cache behavior and memory traffic (especially MergeSort moves).