# Nurlan Ramazan SE-2431

# 1. Introduction & Algorithms

This project implements four classical divide-and-conquer algorithms:  
- MergeSort: reusable buffer, cutoff to insertion sort (Θ(n log n)).  
- QuickSort: randomized pivot, recursion into smaller part (Θ(n log n) on average).  
- Deterministic Select: median of medians (Θ(n)).  
- Closest Pair of Points: divide-and-conquer with strip check (Θ(n log n)).

# 2. Metrics & Testing

Metrics recorded in metrics.csv:  
- algorithm (name)  
- n (input size)  
- time\_ns (execution time)  
- maxDepth (recursion depth)  
  
Validation:  
• Sorting algorithms checked against Arrays.sort.  
• Select checked against sorted k-th element.  
• Closest Pair verified with O(n²) brute force.

# 3. Results

Experimental results confirm theoretical complexities:  
- MergeSort & QuickSort scale ~ n log n.  
- Deterministic Select grows linearly.  
- Closest Pair matches Θ(n log n).  
  
Plots (time\_vs\_n.png, depth\_vs\_n.png) show growth of execution time and recursion depth.

# 4. Workflow & Conclusion

GitHub workflow:  
- main branch for stable releases (v0.1, v1.0).  
- features: mergesort, quicksort, select, closest, metrics.  
  
Conclusion: All algorithms were implemented, metrics logged, and results validated. The project meets assignment requirements.