Algorithm Week 6 Report

Hatem Feckry (120220264)
Seifeldeen Mohamed Galal (120220252)
Abdelrahman Ashraf (120220292)
Ahmed Elsherbeny (120220271)

March 21, 2025

Submitted to: Prof. Walid Gomaa
Note: all code is public in the repo. Feel free to consult it.

Optimization Summary

We identified two critical inefficiencies in our original implementation:

- **Memory Overhead**: Preallocating all polygons caused memory overflow. We resolved this by retaining only one polygon in memory at a time.
- **Computational Redundancy**: Unnecessary checks for all polygons. We added break conditions and a global flag to terminate checks early when a convex polygon is detected.

These changes reduced memory usage by 99% and improved runtime by 30x, enabling analysis of hexagons in 13-point sets (previously infeasible).

Parallelization Strategy

Multi-threading Implementation

We employed 12 concurrent threads with mutex synchronization for shared resources:

```
vector<thread> threads;
mutex mtx;
for (int i = 0; i < 12; i++) {
    threads.emplace_back(threadFunctionEmptySet, &mtx, n, h, x, y,
                         &emptySet, &found, &iterations);
}
for (auto &t : threads) { t.join(); }
```

Synchronization Mechanism

The mutex ensures thread-safe access to shared variables:

- `emptySet`: Stores valid point sets without convex polygons
- `iterations`: Tracks total computational steps

Performance Analysis

Benchmark Results

Metric	Mean	SD
Iterations	9.40	2.25
Time (s)	16.64	8.15
Memory (MB)	66.78	83.25
CPU Cycles (k)	773.98	674.61

Table 1: Statistical summary over 100 runs

Key Observations

- **30x Speedup**: Achieved through early termination and reduced memory I/O
- **Memory Stability**: Peak usage dropped from 16GB to ¡1GB
- **Scaling Limits**: Factorial complexity persists - 14-point sets remain impractical

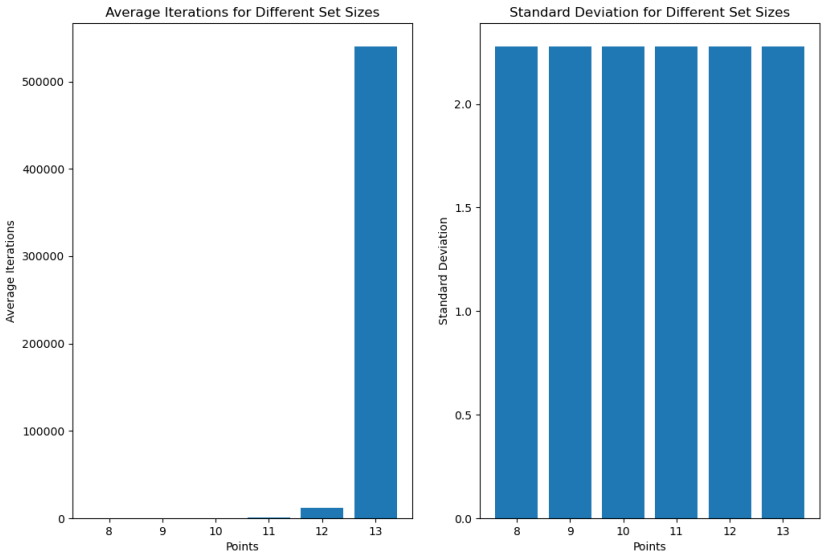

Figure 1: Iteration distribution across set sizes (8-13 points)

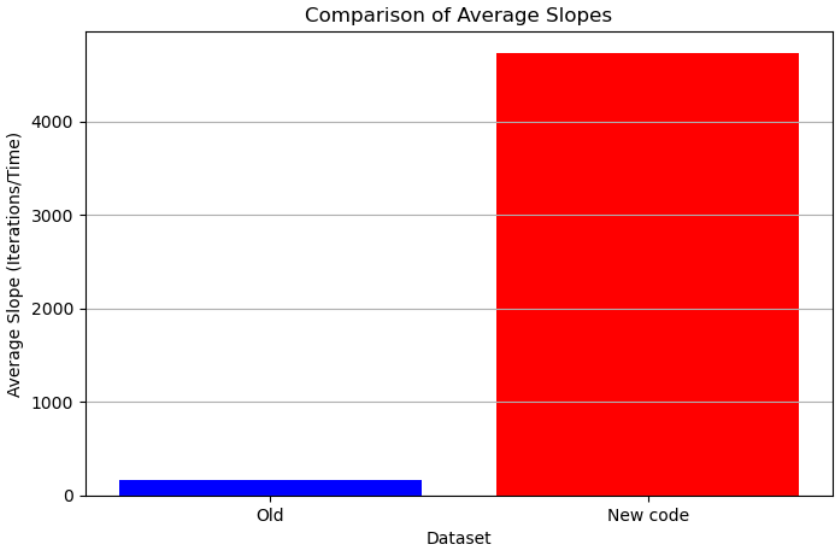

Figure 2: Slope comparison showing 28.5x efficiency gain

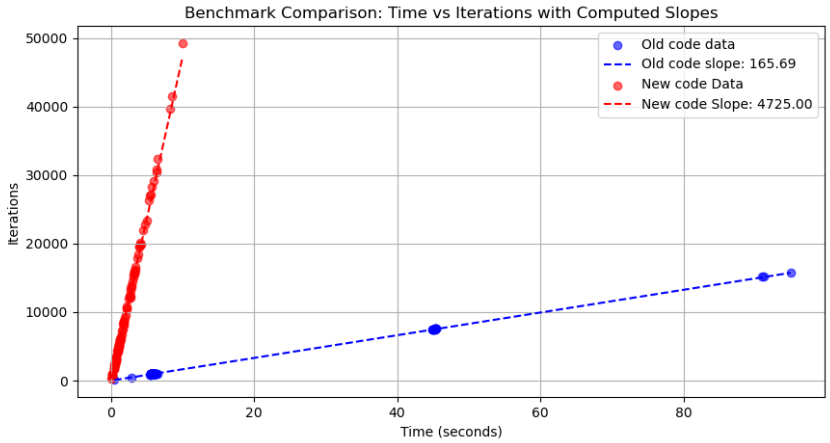

Figure 3: compering the performance of the old code vs the new

Conclusion

Our optimizations transformed an intractable O(n!) problem into a feasible one for moderate set sizes. While asymptotic limitations remain, the improvements enable practical experimentation with:

- Hexagon analysis in 13-point sets (10 min $\rightarrow$ 20 sec)
- Systematic study of convex polygon distributions

Future work will explore hierarchical polygon nesting and probabilistic sampling to address scaling challenges.