reflection

a. stack implementation - python list
   - push (append): O(1) amortized - python lists are dynamic arrays that need to resize but append operations are O(1) on average
   - pop: O(1) - removing the last element doesnt need to move anything else

b. queue implementation - collections.deque
   - push (append): O(1) - deques are implemented as doubly-linked lists with O(1) tail insertion
   - pop (popLeft): O(1) - deques keep O(1) head removal with pointer manipulation

c. priorityQueue implementation - heapq
   - insert: O(log n) - heap insertion needs at most one bubble-up operation for eah level
   - removeMin: O(log n) - heap extraction needs a bubble-down operation after swapping with the last element

experimental observations

algorithm comparison

path length

- BFS and A*: consistently found the shortest paths — equal length
- DFS: found path 1.5-3x longer because of deep exploration before backtracking

path visualization:

- BFS/A: smooth, direct routes with minimal turns
- DFS: zig-zag patterns with visible backtracking

cells explored:

Algorithm	Avg. Cells Explored
DFS	350-500
BFS	450-600
A*	120-250

Example (Seed 8675309):

- DFS: 412 cells, path length 48
- BFS: 587 cells, path length 32
- A*: 183 cells, path length 32

key comparisons:

variations:

1. path optimality:
    - BFS/A* guarantees the shortest path in unweighted grids
    - DFS results in deeper exploration — finding suboptimal paths
2. exploration patterns:
    - BFS expands uniformaly like a wave
    - A* focuses exploration toward the goal with heuristic
    - DFS makes tunnels until it hits a obstacle

similarities:

- all algorithms eventually find a path if it exists
- parent-point structure allows identical path reconstructions

efficiency analysis:

most efficient: A*

- heuristic guidance reduces explored cells by 60/70% vs BFS
- keeps optimality while avoiding BFS uniform expansion
- priority queue O(log n) outweighs the cost of extra explored cells in BFS and DFS

maze variability:

- results were consistent across seeds
- A* showed strongest consistency in path length/cell count
- DFS varied most based on initial branching decision

unexpected:

A* sometimes explored more cells than BFS in open areas where manhattan distance overestimated the true cost — kept path optimality. shows heuristic role in balancing exploration vs exploitation