EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

AING216 Basic Search Strategies

LAB III

Question 1. A* on the weighted graph

Use the same graph and heuristic table style as in Lab II.

From	To	Cost
A	B	1
A	C	3
B	D	1
B	E	5
C	F	2
D	G	3
E	G	1
F	G	2

A	B	C	D	E	F	G
h=6	h=4	h=4	h=2	h=1	h=2	h=0

- Show the initial OPEN list after expanding A.
- List the node expansion order until the goal is selected.
- Give the final shortest path and the total path cost.
- State whether the given heuristic is admissible and consistent for this graph.

Question 2. A* on the maze

Maze encoding: 0 = free cell, 1 = obstacle, S = start, G = goal.

S	0	0	0
1	1	0	1
0	0	0	0
0	1	1	G

- Use Manhattan distance h(n) = |x1 - x2| + |y1 - y2|.
- List the first four expanded cells in order.
- Write the final path from S to G as coordinates.
- Give the number of moves in the solution.
- In one sentence, explain why Manhattan distance fits this grid.

Question 3. Iterative A* threshold tracing

Run IDA left-to-right on the following search tree. The start node has h(S) = 4. Use the child order A before B, and inside B use E before F.*

Parent	Child	Step cost	h(child)
S	A	1	4
A	C	2	2
C	G	3	0
S	B	2	2
B	E	2	3
B	F	1	2
F	G	3	0

- What is the initial f-limit?
- What threshold sequence does IDA* use before the goal is found?
- Which goal path is returned first under left-to-right expansion, and what is its total cost?
- Why does IDA* usually revisit nodes more than A*?

Question 4. Choosing between A* and Iterative A*

Suppose you must solve a 50 x 50 four-direction grid with unit step cost and Manhattan distance. RAM is tight: you can store only a few thousand states before the program slows badly.

- Which algorithm is safer from memory exhaustion: A* or IDA*? Explain briefly.
- Which algorithm usually performs fewer re-expansions when the heuristic is decent?
- If the heuristic is weak, which algorithm is more likely to suffer in runtime? Why?
- Give one practical situation where A* is the better choice, and one where IDA* is the better choice.

Question 5. SMA* with a memory limit

Assume SMA may keep at most 4 leaf nodes in memory. Use the tie-break rule: if several leaves have the same worst f-value, delete the deepest rightmost one.*

Parent	Child	Step cost	h(child)
S	A	2	4
A	D	2	3
A	E	2	1
E	H (goal)	1	0
S	B	1	6
S	C	3	2
C	F	2	2
C	G	4	0

Current leaves after expanding S, then C, then A: B(f=7), F(f=7), G(f=7), D(f=7), E(f=5)

- Which leaf is forgotten first?
- What backed-up value is stored in its parent?
- Which leaf will SMA* expand next?
- Why can the forgotten branch still be regenerated later if it becomes promising again?