INTRODUCTION TO ARTIFICIAL INTELLIGENCE
Lab 7 – Search Algorithms – A*

A* – a brief reminder

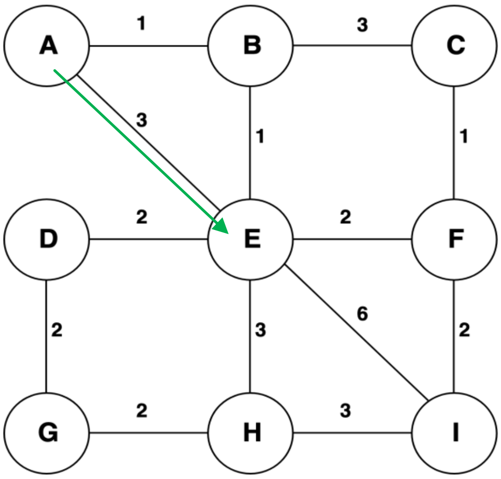
A* is a search algorithm, i.e., an algorithm for finding the shortest path between two nodes in a graph. Apart from costs associated with graph edges, A* requires having estimated distances from each node to the target. During the search process, A* selects the next node to visit so that the total cost of the already traversed path and the heuristic costs associated with the path to the target is minimal. If the used heuristic function is admissible, i.e., does not overestimate the costs, then A* is optimal and complete.

1) **A* algorithm:** Find the shortest path between nodes A (source) and I (target) in the non-directed graph. The distance (cost) between two nodes is written next to the edge connecting these nodes. After how many iterations the A* algorithm has found the shortest path between A and I?

Heuristics/costs costs from node (x) to I

	A	B	C	D	E	F	G	H	I
h(x)	8	7	6	5	4	3	2	1	0

I iteration

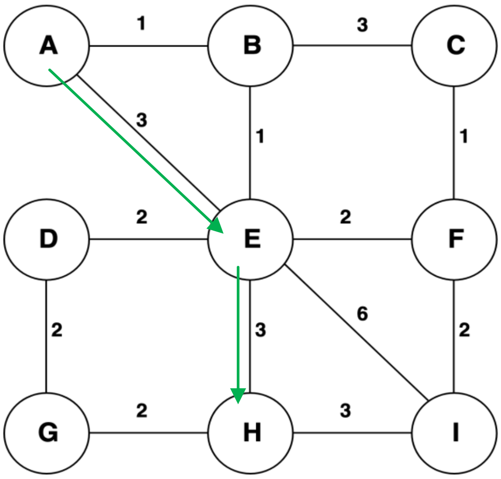


Closed set: A
Open set: E B

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→E	3	4	7
A→B	1	7	8

Paths are kept sorted for convenience

II iteration



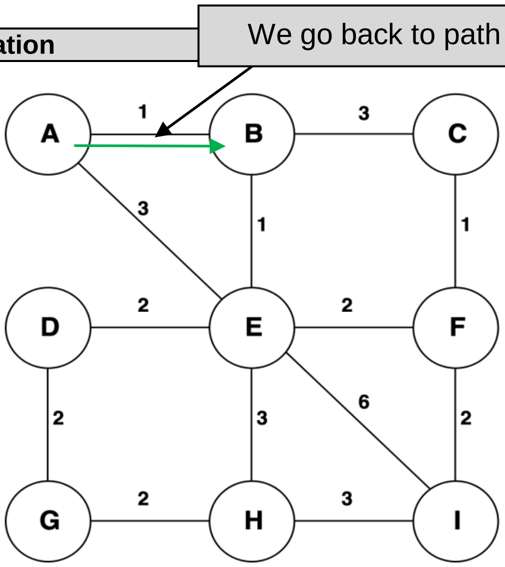
Closed set: A E
Open set: H B F I D

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→E→H	6	1	7
A→B	1	7	8
A→E→F	5	3	8
A→E→I	9	0	9
A→E→D	5	5	10

A→E→B does not have to be added because a shorter path from A to B has already been found (A→B).

III iteration

We go back to path A→B

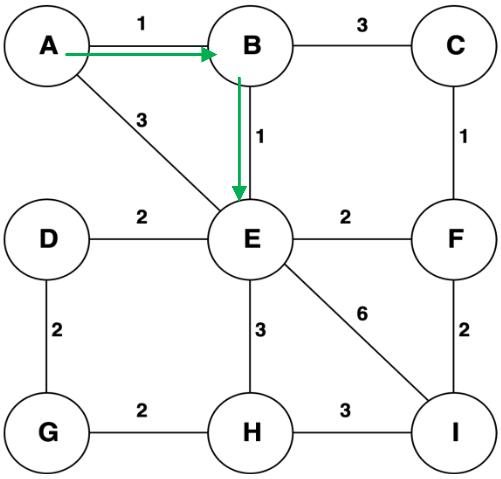


Closed set: A E H
Open set: B F I D G

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→B	1	7	8
A→E→F	5	3	8
A→E→I	9	0	9
A→E→D	5	5	10
A→E→H→G	8	2	10

A→E→H→I does not have to be added there already exist a path of the same length (A→E→I).

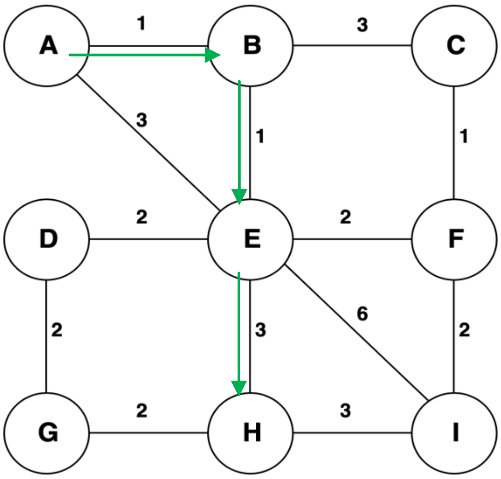
IV iteration



Closed set: A E H B
Open set: E F I D G C

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→B→E	2	4	6
A→E→F	5	3	8
A→E→I	9	0	9
A→E→D	5	5	10
A→E→H→G	8	2	10
A→B→C	4	6	10

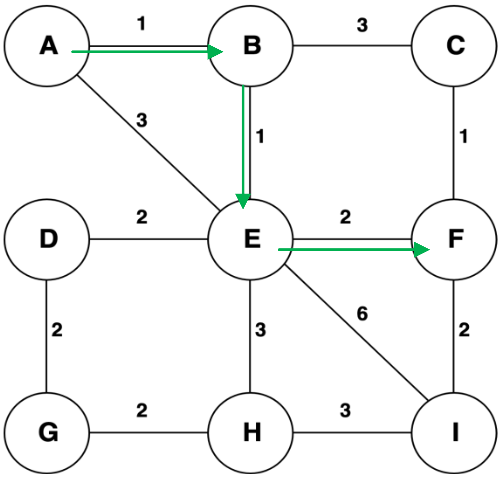
V iteration



Closed set: A E H B
Open set: H F I D C G

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→B→E→H	5	1	6
A→B→E→F	4	3	7
A→B→E→I	8	0	8
A→B→E→D	4	5	9
A→B→C	4	6	10
A→E→H→G	8	2	10

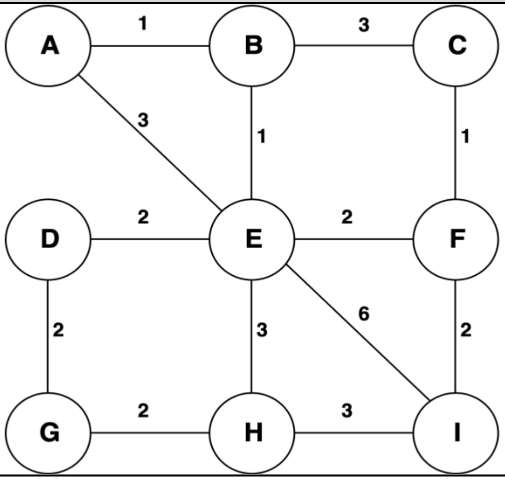
VI iteration



Closed set: A E H B
Open set: F I D G C

Path	g(x)	h(x)	f(x)=g(x)+h(x)
A→B→E→F	4	3	7
A→B→E→I	8	0	8
A→B→E→D	4	5	9
A→B→E→H→G	7	2	9
A→B→C	4	6	10

VII iteration



Closed set: A E H B F
Open set: I D G C

Path	g(x)	+h(x)	f(x)=g(x)
A→B→E→F→I	6	0	6
A→B→E→D	4	5	9
A→B→E→H→G	7	2	9
A→B→C	4	6	10

THE SHORTEST PART HAS BEEN FOUND