# A* Shortest Path Algorithm

The A star algorithm is similar to Dijkstra's algorithm but uses heuristics to find a shorter path than the original algorithm. It labels each vertex or node with a value equal to its straight line distance. This is an estimate of the distance to get to each point and it can use this to try to minimise the distance. This is called heuristics as it is basing its solution on an estimate which may not be correct. It will always use the lowest heuristic value as it needs to be sure it does not overestimate. This is why it uses the straight line value as you can't get from one point to the next shorter than a straight line. It minimises the distance to the end meaning that it ignores paths that are longer

How it Works

Step 1)start at node A and label every node which can be directly reached from A with a value if the straight line distance to it plus its heuristic distance from the end of the graph.

2) Label A as closed and mode onto the next open node with the smallest value and mark this as the current node. Calculate the distances for every node that can be reached for the current node and repeat the steps until you reach the target node.

An example
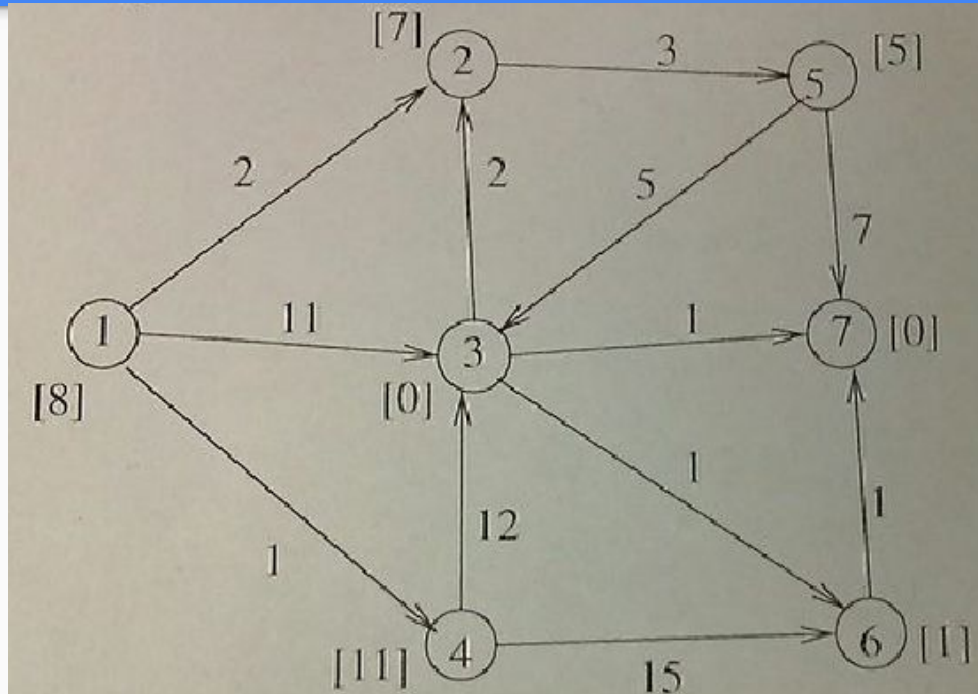
An Example

Node(n)	Path distance(g)	Heuristic distance(h)	f=g(n)+h(n)	Previous node
1	0	8		
2		7		
3		0		
4		11		
5		5		
6		1		
7		0		

An Example

Node(n)	Path distance(g)	Heuristic distance(h)	f=g(n)+h(n)	Previous node
1	0	8	8	
2	2	7	9	1
3	11	0	11	1
4	1	11	12	1
5		5		
6		1		
7		0		

An Example

Node(n)	Path distance(g)	Heuristic distance(h)	f=g(n)+h(n)	Previous node
1	0	8	8	
2	2	7	9	1
3	11	0	11	1
4	1	11	12	1
5	3+2=5	5	10	2
6		1		
7		0		

An Example

Node(n)	Path distance(g)	Heuristic distance(h)	f=g(n)+h(n)	Previous node
1	0	8	8	
2	2	7	9	1
3	11, 5+5=10	0	10	5
4	1	11	12	1
5	3+2=5	5	10	2
6		1		
7	5+7=12	0	12	5

An Example

Node(n)	Path distance(g)	Heuristic distance(h)	f=g(n)+h(n)	Previous node
1	0	8	8	
2	2	7	9	1
3	11, 5+5=10	0	10	5
4	1	11	12	1
5	3+2=5	5	10	2
6		1		
7	5+7=12	0	12	5

An example

The shortest path is 1, 2, 5