Common heuristic functions:

1) Euclidean distance: $H(x_n, y_n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$

2) Manhattan distance: $H(x_n, y_n) = |x_n - x_g| + |y_n - y_g|$

A* algorithm – pseudo code

- For each node n in the graph
  - n.f = Infinity, n.g = Infinity
- Create an empty list.
- start.g = 0, start.f = H(start) add start to list.
- While list not empty
  - Let current = node in the list with the smallest f value, remove current from list
  - If (current == goal node) report success
  - For each node, n that is adjacent to current

    If (n.g > (current.g + cost of edge from n to current))

    n.g = current.g + cost of edge from n to current

    n.f = n.g + H(n)

    n.parent = current

    add n to list if it isn't there already