**Algorithm 1** A* Search Algorithm (Graph)

```
function A*(start, goal)
    closedset ← the empty set
    openset ← start
    came_from ← the empty map[string]

    g_score[start] ← 0
    f_score[start] ← g_score[start] + heuristic_cost_estimate(start, goal)

    while openset ≠ ∅ do
        current ← the node in openset having the lowest f_score[] value

        if current = goal then return reconstruct_path(came_from, goal)
        remove current from openset
        add current to closedset

        for all neighbour ∈ neighbour_nodes(current) do
            if neighbour ∈ closedset then
                continue
            neighbour_g_score ← g_score[current] + dist_between(current, neighbour)
            if neighbour not in openset || neighbour_g_score < g_score[neighbour] then
                came_from[neighbour] ← current
                g_score[neighbour] ← neighbour_g_score
                f_score[neighbour] ← g_score[neighbour] + heuristic_cost_estimate(neighbour, goal)
                if neighbour ∉ openset then
                    addneighbourtoopenset
    return failure
```