A* Algorithm 最佳性證明（反證法）

A* algorithm 是一種最佳優先搜尋（best-first search）方法，常用於解決最佳化問題，可以找出解答空間搜尋樹的根節點到目標節點的最小權重加總。對於任意節點 n，A* 使用以下評估函數：

$$\boldsymbol{f(n) = g(n) + h(n)}$$

其中 $g(n)$ 表示從起點(或搜尋樹的根節點)到節點 n 的實際成本；$h(n)$ 表示從節點 n 到目標節點的估計未來成本；$h^*(n)$ 表示真正的最小未來成本。

A* algorithm 必須滿足可接受啟發函數(admissible heuristic) 條件，也就是

$$\boldsymbol{h(n) \leq h^*(n)}$$

因此可得：

$$\boldsymbol{f(n) = g(n) + h(n) \leq g(n) + h^*(n) = f^*(n)}$$

亦即 A* 對總成本的估計值不會高於真正的最佳成本。以下利用反證法證明：**<u>A* algorithm 第一次找到(或拜訪)的目標節點一定是最佳解</u>。**

A* 最佳性證明：反證法

設目前被 A* algorithm 選取的節點 x 為目標節點。因為 x 已經是目標，因此其未來的評估成本與真實成本均為 0。我們可得:

$$\boldsymbol{h(x) = 0 \text{ 且 } h^*(x) = 0}$$

因此：

$$\boldsymbol{f^*(x) = g(x) + h^*(x) = g(x) + h(x)= f(x)}$$

接著假設 x 並不是最佳解。若此假設成立，則一定存在另一條由起點(或根節點)到目標節點的真正最佳路徑，其總成本比 x 更小。

在該最佳路徑上，必定存在某個節點 y，此節點尚未被選取，但已經存在於 priority queue 中。由於該路徑比 x 更佳，因此：

$$\boldsymbol{f^*(y) < f^*(x)}$$

然而，A* algorithm 每一次都會從 priority queue 中選擇 f 值最小的節點進行擴展。既然 x 已被選取，而 y 尚未被選取，因此必定有：

$$\boldsymbol{f(x) \leq f(y)}$$

另一方面，由 admissible heuristic 條件可得：

$$\boldsymbol{f(y) \le f^*(y)}$$

再加上先前已經證明：

$$\boldsymbol{f^*(x) = f(x)}$$

因此可以得到：

$$\boldsymbol{f^*(x) = f(x) \le f(y) \le f^*(y)}$$

也就是：

$$\boldsymbol{f^*(x) \le f^*(y)}$$

但這與前面假設的 $\boldsymbol{f^*(y) < f^*(x)}$ 互相矛盾，故原本「x 不是最佳解」的假設不成立。因此，A* algorithm 第一次找到的目標節點一定是最佳解。