WEEK1

爬山演算法:

是一種[局部擇優](https://zh.wikipedia.org/w/index.php?title=%E5%B1%80%E9%83%A8%E6%8B%A9%E4%BC%98&action=edit&redlink=1)的方法，採用啟發式方法，是對[深度優先搜尋](https://zh.wikipedia.org/wiki/%E6%B7%B1%E5%BA%A6%E4%BC%98%E5%85%88%E6%90%9C%E7%B4%A2)的一種改進，它利用[回饋](https://zh.wikipedia.org/wiki/%E5%8F%8D%E9%A6%88)[資訊](https://zh.wikipedia.org/wiki/%E4%BF%A1%E6%81%AF)幫助生成解的決策。

爬山演算法一般存在以下問題：

1. 局部最大
2. 高地：也稱為[平頂](https://zh.wikipedia.org/wiki/%E5%B9%B3%E9%A0%82)，搜尋一旦到達高地，就無法確定搜尋最佳方向，會產生[隨機](https://zh.wikipedia.org/wiki/%E9%9A%8F%E6%9C%BA)走動，使得搜尋效率降低。
3. 山脊：搜尋可能會在[山脊](https://zh.wikipedia.org/wiki/%E5%B1%B1%E8%84%8A)的兩面來回震盪，前進步伐很小。

解決方法：**隨機重新啟動爬山演算法**

參考資料:維基百科

https://zh.wikipedia.org/wiki/%E7%88%AC%E5%B1%B1%E7%AE%97%E6%B3%95