

## Multistage graph by A\* algorithm

----- ↓ 跟題目無關，但重要 ↓ -----

A\*演算法的觀念在於時時刻刻都選擇評估函數值最小的那個選項

而評估函數為  $f(n) = g(n) + h(n)$

$g(n)$ 為目前已經花費的成本

$h(n)$ 是未來預估的成本

A\*能否有效的執行關鍵就在於  $h(n)$ 的定義方式

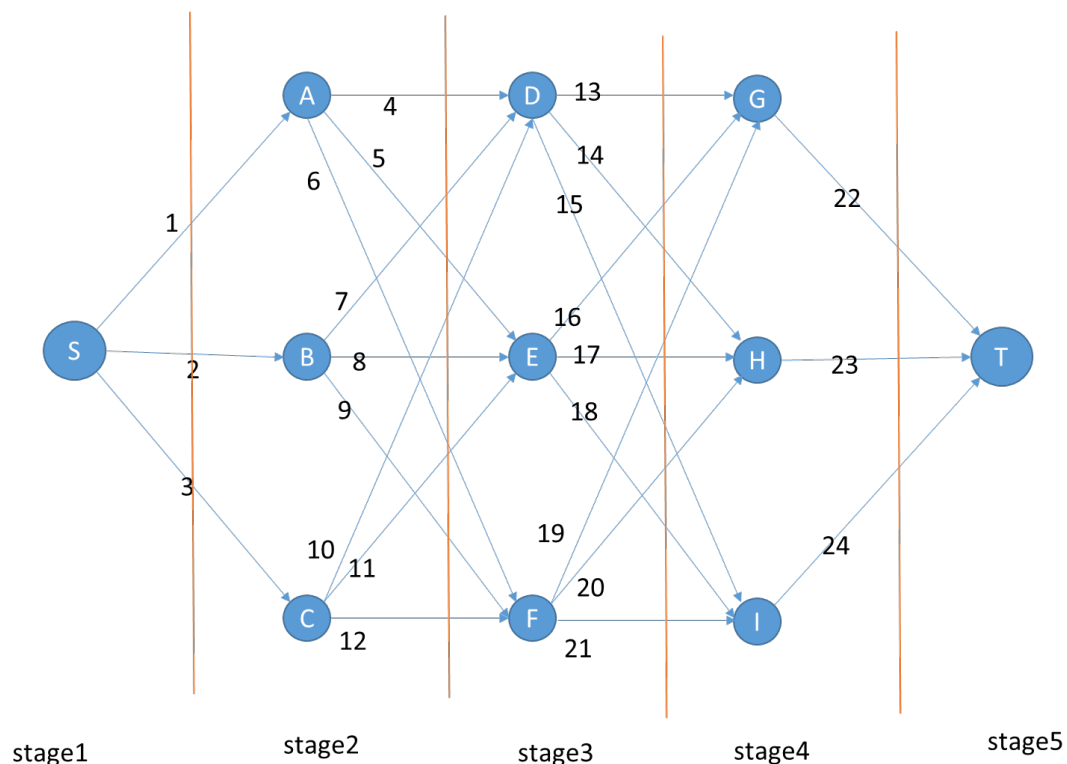
因為沒人能夠真正預知到未來所花費的成本為多少,所以我們只能用一些方式去猜測,只要猜得夠接近真實的成本,那 A\*就能發揮它的長處,如果定義的不好,不夠接近真實的成本,則 A\*的效果就會非常差,所以 A\*的關鍵就在於  $h(n)$ 要猜得夠準

在老師的投影片中,因為應用的問題是 multistage graph,我們能取得的未來資訊就只有到達下一個節點所需的成本,所以我們的  $h(n)$ 只能定義成下個節點的成本,這是無可奈何的定義方式,如果能夠取得更多資訊的話則最好是用別的方式去定義  $h(n)$ ,出來的結果會更準確

----- ↑ 跟題目無關，但重要 ↑ -----

請實作 A\*演算法,評估函數與老師上課投影片中的定義方式相同

Input 的 graph 格式固定為以下的圖片所示



總共 11 個點 24 條邊

測資會依序給出編號為 1~24 的邊的成本

最後會指定一個 **stage**,請輸出從 **S** 點至該 **Stage** 為止,最低的  $f(n)$  為多少  
例如,從 **S** 到 **stage3** 的話則必須考慮從 **S** 到 **D/E/F** 與 **D/E/F** 各自延伸出去的所有  
邊的成本總和

Sample input

1  
2  
3  
6  
5  
4  
7  
8  
9  
12  
11  
10  
13  
14  
15  
18  
17  
16  
19  
20  
21  
24  
23  
22  
3

Sample output

20