

# 大學程式設計先修檢測

2018.06.10

## 實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或.py，如下表所示。另若以 Java 撰寫程式，class 名稱必須與檔名（P1, P2, P3, 或P4）一致。

	C	C++	Java	Python
特殊編碼	P1.c	P1.cpp	P1.java	P1.py
完全奇數	P2.c	P2.cpp	P2.java	P2.py
工作排程	P3.c	P3.cpp	P3.java	P3.py
反序數量	P4.c	P4.cpp	P4.java	P4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以0分計算。
3. 程式內不應有開檔、讀檔、寫檔等程序，資料讀取皆應來自標準輸出入，資料輸出入格式詳各題說明。評分時，系統將自動導入測試資料，每行輸入資料最後一定有換行（UNIX 格式）。

### 第 3 題 工作排程

#### 問題描述

有  $M$  個工作要在  $N$  台機器上加工，每個工作  $i$  包含若干個工序  $o_{ij}$ ，這些工序必須依序加工，也就是前一道工序  $o_{i(j-1)}$  完成後才可以開始下一道工序  $o_{ij}$ 。每一道工序  $o_{ij}$  可以用一個有序對  $(k_{ij}, t_{ij})$  來表示它需要在機器  $k_{ij}$  上面花費  $t_{ij}$  小時來完成。每台機器一次只能處理一道工序。

所謂一道工序  $o_{ij}$  的「最早完成時間  $c_{ij}^*$ 」是指考慮目前排程中機器  $k_{ij}$  之可用性以及前一道工序  $o_{i(j-1)}$ （如果該工序存在）之完成時間後可得的最早完成時間。工廠經理安排所有工序的排程規則如下：

針對每一個工作的第一個尚未排程的工序，計算出此工序的「最早完成時間」，然後挑選出最早完成時間最小的工序納入排程，如果有多個最早完成時間都是最小，則挑選其中工作編號最小之工序。一個工序一旦納入排程就不會再更改，重複以上步驟直到所有工序皆納入排程。

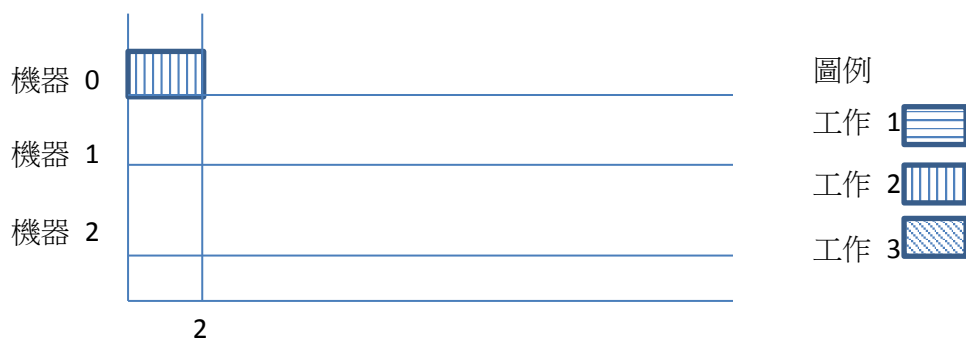
我們總是從時間 0 開始排程，每個工作的完成時間為其最後一個工序的完成時間，本題的目標是計算出每個工作的完成時間並輸出其總和。

以下以一個例子來說明，在此例中，有三個工作要在三台機器上排程，各工作的資料如下。

	工序	說明
工作 1	$o_{11} = (2, 4)$ $o_{12} = (1, 1)$	此工作有兩道工序，第一道需要在機器 2 執行 4 小時，第二道需要在機器 1 執行 1 小時。
工作 2	$o_{21} = (0, 2)$ $o_{22} = (2, 2)$ $o_{23} = (0, 1)$	有三道工序，第一道需要在機器 0 執行 2 小時，餘類推。
工作 3	$o_{31} = (0, 7)$	有一道工序需要在機器 0 執行 7 小時。

排程的過程說明如下：

1. 在開始時，每個工作都是考慮第一道工序，三個工作第 1 道工序需要的時間分別是  $t_{11} = 4$ 、 $t_{21} = 2$ 、 $t_{31} = 7$ ，這也是它們的最早完成時間，也就是  $c_{11}^* = 4$ 、 $c_{21}^* = 2$ 、 $c_{31}^* = 7$ ，因此會先排  $o_{21}$ 。



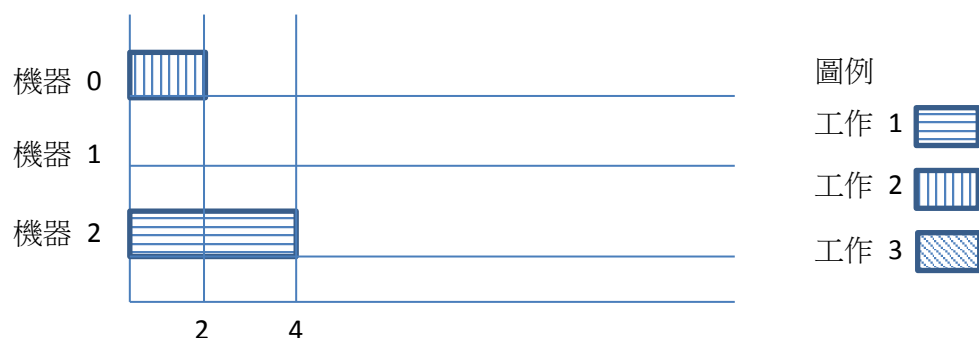
2. 接下來，三個工作要考慮的工序分別是第 1、2、1 個工序，即  $o_{11}$ 、 $o_{22}$  和  $o_{31}$ 。

(1)  $o_{11}$  需要機器 2 執行 4 小時，而機器 2 可以開始加工的時間點是 0； $o_{11}$  沒有前一道工序。因此，這工序可以開始的時間是  $\max(0, 0) = 0$ 。是故，其最早完成時間  $c_{11}^* = \max(0, 0) + 4 = 4$ 。

(2)  $o_{22}$  需要機器 2 執行 2 小時，而機器 2 可以開始加工的時間點是 0； $o_{22}$  前一道工序  $o_{21}$  的完成時間是 2。因此，這工序可以開始的時間是  $\max(0, 2) = 2$ 。是故，其最早完成時間  $c_{22}^* = \max(0, 2) + 2 = 4$ 。

(3)  $o_{31}$  需要機器 0 執行 7 小時，而機器 0 可以開始加工的時間點是 2； $o_{31}$  沒有前一道工序。因此，這工序可以開始的時間是  $\max(2, 0) = 2$ 。是故，其最早完成時間  $c_{31}^* = \max(2, 0) + 7 = 9$ 。

因此，由於  $c_{11}^*$  和  $c_{22}^*$  都是最小，根據規則，工作編號小的先排，所以會排  $o_{11}$ 。

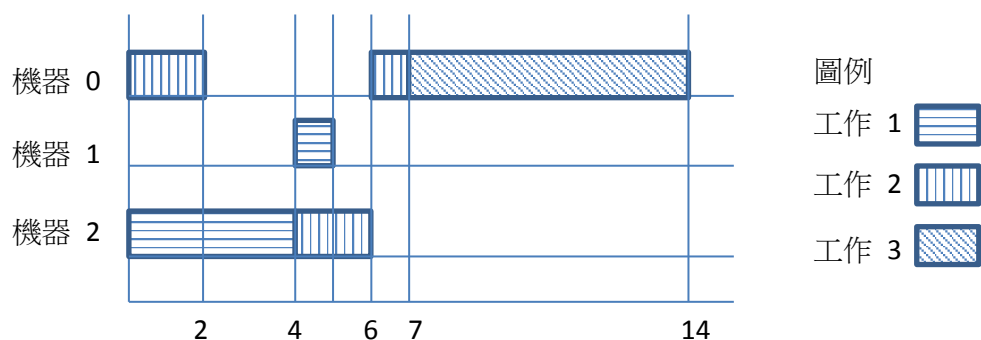


3. 三個工作目前要考慮的工序分別第 2、2、1 個工序。依照類似的推論，我們可以得到  $c_{12}^* = 5$ ， $c_{22}^* = 6$ ， $c_{31}^* = 9$ ，因此排  $o_{12}$ 。工作 1 的工序均已排完，所以它的完成時間是 5。

4. 剩下工作 2 與 3。 $c_{22}^* = 6$ ， $c_{31}^* = 9$ ，因此先排  $o_{22}$ 。

5.  $c_{23}^* = 7$  而  $c_{31}^* = 9$ ，因此排  $o_{23}$ ，工作 2 的工序已排完，所以它的完成時間是 7。

6. 剩下工作 3，因為機器 0 的下一個可以開始時間是 7， $o_{31}$  的完成時間是  $7+7=14$ 。



三個工作的完成時間分別是 5、7、14，所以最後輸出答案  $5+7+14=26$ 。

### 輸入格式

第一行有兩個整數  $N$  與  $M$ ，代表  $N$  台機器與  $M$  個工作，接下來有  $M$  個工作的資訊，輸入的順序即是工作編號順序。每個工作資訊包含兩行，第一行是整數  $P$ ，代表到工序數量；第二行是  $2 \cdot P$  個整數，每兩個一組依序代表一道工序的機器編號與需求時間。機器的編號由 0 開始。參數  $N$ 、 $M$ 、 $P$  以及每個工序的需求時間都是不超過 100 的正整數。

## 輸出格式

輸出每個工作的完成時間的總和。

### 範例一：輸入

```
3 3
2
2 4 1 1
3
0 2 2 2 0 1
1
0 7
```

### 範例一：正確輸出

26

### 範例二：輸入

```
2 3
1
0 4
1
1 5
1
1 3
```

### 範例二：正確輸出

15

## 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測試筆數給分。其中：

第 1 子題組 20 分：只有一台機器，各工作只有一道工序。

第 2 子題組 30 分：各工作只有一道工序。

第 3 子題組 50 分：無其他限制。