
程式目的

- 這是一個結合實際應用的程式。給定一個數位邏輯電路，找出 Critical Path。

名詞說明

- Gate：在數位邏輯電路中的一個單元，代表一個邏輯運算。例如 AND Gate（請見附圖 AND.png）

會接受兩個 input 訊號，產生一個 output 訊號。其中 output 訊號會是兩個 input 訊號做完

AND 的結果。

- Gate Input：Gate 的 input 訊號，不同種類的 Gate 可能有不同個數的 input，例如 NOT Gate

（請見附圖 NOT.png）會有 1 個 input；但 AND Gate 會有 2 個 input。

- Gate Output：Gate 的 output 訊號，不同種類的 Gate 可能有不同個數的 output，但在本次

作業中使用的 Gate 皆只有 1 個 output。

- Gate Delay：Gate 需要時間來處理訊號，從訊號 input 到訊號 output 達到穩定狀態之間的時

間稱為 Gate Delay。不同種類的 Gate 之 Gate Delay 可能會不同。例如 NOT Gate 的 Gate

Delay 可能比 AND Gate 的 Gate Delay 來得短。

- Circuit：一個 Circuit 可用來實現一群複雜的邏輯運算，一個 Circuit 可包含數個 input 訊

號、數個 output 訊號，與數個 Gate，某一個 Gate 的 output 訊號可成為另一個 Gate 的

input 訊號，由此將數個 Gate 串接以達成一連串的邏輯運算（請見附圖 Circuit.png）。

- Critical Path：從 Circuit 的 input 訊號到 output 訊號間，可能形成數條路徑，在這些路徑

中，擁有最大 Gate Delay 總和的路徑，稱為 Critical Path。一個 Circuit 可能擁有數條

Critical Path。就意義上而言，Critical Path 決定了這整個 Circuit 的反應時間。

程式需求

-
- 程式必需命名為 AOV.exe。
 - 程式必需是自己寫的。
 - 程式必需以 C 語言完成，並能以 VC6.0 以上或 gcc 編譯成功。
 - 程式必需加入適當的註解，並維持良好的寫碼風格。
 - 程式執行後必需讀入輸入檔，由輸入檔中的 Gate 與 Circuit 資訊，找出 Critical Path

並將結果寫至輸出檔。請見下方「輸入檔格式」與「輸出檔格式」。

- 執行程式時不需由 command line 給任何參數。
- 程式在完成輸出檔後，應立即結束，請勿使用 system("pause") 之類的東西讓畫面暫停。
- 不需要做任何容錯處理，例如可以假設輸入檔格式完全正確，可以假設檔案讀寫不會發生例外狀況。
- 程式至少要能通過助教提供的 Test Case，請見下方「驗證與測試」。

輸入檔格式

- 共有兩個輸入檔，一個檔名為 Gates.txt，另一個檔名為 Circuit.txt。
- 輸入檔與 AOV.exe 在同一目錄下。
- 所有輸入檔都遵守以下規則：
 - (1) 每一列皆以 \r\n 換行。
 - (2) 各「欄位」間以一個空白字元分隔。
 - (3) 除了欄位分隔外，檔案中不會有任何空白字元或 \t 字元。
 - (4) 不會有空白列。
 - (5) 所有的「數字」皆是正整數。
 - (6) 所有的「名稱」皆是不含空格的字串，字串由英文大小寫字元與數字字元組成，大小寫有別。
 - (7) 所有的「名稱」皆不超過 5 個字元。
- Gates.txt 中存放各種類 Gate 的定義與特性，其檔案格式如下：
 - (1) 第 0 列恰包含一個正整數，代表要定義的 Gate 種類個數。
 - (2) 從第 1 列開始，每一種類的 Gate 定義佔一列，每一列包含 3 個欄位。
 - 欄位 0 是種類名稱。
 - 欄位 1 是 input 個數。
 - 欄位 2 是 Gate Delay。

注意：不同列間的種類名稱不會重覆。本次作業中所有 Gate 的 output 個數皆為 1。

- Gates.txt 輸入檔範例如下：

```
3
NOT 1 1
NOR 2 1
NAND 2 2
```

- Circuit.txt 中存放 Circuit 的描述（其所描述的必是正確的 Circuit，且 Circuit 中不會有

loop 存在，也就是某個 Gate 的 output 不會再度成為自己的 input），其檔案格式如下：

- (1) 第 0 列包含 3 個欄位。

欄位 0 是整個 Circuit 的 input 個數。

欄位 1 是整個 Circuit 的 output 個數。

欄位 2 是 Gate 個數。

- (2) 第 1 列會列出整個 Circuit 的 input 訊號名稱，每個名稱佔用一個欄位。

- (3) 第 2 列會列出整個 Circuit 的 output 訊號名稱，每個名稱佔用一個欄位。

- (4) 從第 3 列開始，每個 Gate 佔一行，假設該 Gate 有 n 個 input，則該行會有 $n+2$ 個欄位。

欄位 0 是 Gate 的 output 訊號名稱。

欄位 1 是 Gate 種類名稱，此名稱必與 Gates.txt 中定義的名稱相符。

欄位 2~欄位 $n+1$ 是 Gate 的 input 訊號名稱。

- Circuit.txt 輸入檔範例如下：

```
4 2 6
x2 x3 x4 x5
z1 z2
A NOR x2 x3
B NAND x3 x4
C NOR x5 A
z1 NAND x2 C
D NOR B C
z2 NOT D
```

- 綜合上面給的 Gates.txt 與 Circuit.txt 範例，這個數位邏輯電路應該長成像附圖 Circuit.png 的樣子。

- 注意：附圖 Notice.png 亦是一個合法的 Circuit，注意其中 z1 是 Circuit 的 output

訊號，但

同時也是 AND Gate 的 input 訊號；x3 同時是 Circuit 的 input 訊號與 output 訊號，也是

NOR Gate 的 input 訊號。在測試程式中的一個 Test Case 即是採用類似的 Circuit。

輸出檔格式

- 輸出檔檔名必需為 Output.txt。
- 輸出檔必需與 AOV.exe 在同一目錄下。
- 輸出檔必需遵守以下規則：
 - (1) 每一列皆以 \r\n 換行。
 - (2) 各「欄位」間以一個空白字元分隔。
 - (3) 除了欄位分隔外，檔案中不會有任何空白字元或 \t 字元。
 - (4) 不會有空白列。
 - (5) 所有的「數字」皆以整數表示。
- 輸出檔格式如下：
 - (1) 第 0 列恰包含一個正整數，表示 Critical Path 上的 Gate Delay 總和。
 - (2) 從第 1 列開始，每條 Critical Path 佔一列，Critical Path 上的每個訊號名稱佔一列，
訊號名稱必需與 Circuit.txt 輸入檔中的名稱符合，大小寫有別，列印時必需由
input 訊
號往 output 訊號的方向循序列印。當有 2 條以上的 Critical Path 時，Critical
Path
的列印順序不拘。
注意：假設正確答案是有 n 條 Critical Path，則測試程式在驗證你的輸出檔正確
性時，
只會讀取第 1 列～第 n 列來驗證。

- 綜合上面給的 Gates.txt 與 Circuit.txt 範例，計算得出的輸出檔範例如下：

```
4
x2 A C z1
x3 A C z1
x2 A C D z2
x3 A C D z2
x3 B D z2
x4 B D z2
```

驗證與測試

- 為了讓你方便驗證自己寫的程式有無錯誤，助教提供一個快速驗證程式 Tester.exe，請將

Tester.exe 與你的 AOV.exe 放在同一個目錄下（目錄下只要有這兩個檔即可），然後雙擊兩下

Tester.exe 執行，Tester.exe 會使用數個預先訂好的 Test Case 來驗證你程式的正確性，並

將結果回報給你。

- Tester.exe 只能在 Windows 作業系統上執行。
- 來 Demo 前請務必先使用 Tester.exe 驗證你的程式，並確定你在每個 Test Case 的結果中都

看到「成功!!」字串。

- 你必需自己使用多一點的 Test Case 來驗證你程式的正確性，Demo 時助教會用其他的 Test Case 來測試你的程式，但所有的 Test Case 都會符合此份文件中所描述的規範。

繳交期限與方式

- 01/08(二) 23:55 上傳，1/9(三) 現場 Demo。

評分標準

- 評分項目包含以下項目：程式正確性、註解、寫碼風格、解法的設計（包含程式架構、使用的演

算法、資料結構、時間與空間的效率等等）、助教發問