

# Program#2 作業說明與輸出規範

## Program#2 作業說明與輸出規範

1. 輸出格式注意事項
2. 輸出合法條件（必須同時滿足下列所有條件）
3. 執行資源與限制
4. 作業規則與評分標準
5. evaluator.py 使用方式
  - (1) 執行方式
  - (2) 會自動檢查項目
  - (3) 驗證通過/失敗輸出
6. 常見錯誤提醒
7. 補充提醒

## 1. 輸出格式注意事項

- 所有輸出座標皆為 **1-indexed**（從 1 開始編號）。
- 統計資訊與各 Net 結果需嚴格按照範例如下格式依序輸出（每條 Net 輸出需依照 input 順序）：

```
#interconnections routed = 2
Total interconnection length = 30
The longest interconnection = 2; length = 18
Total number of bends = 3
Interconnection 1: length = 12, #bends = 1
(4, 10), (5, 10), (6, 10), ..., (9, 3)
Interconnection 2: length = 18, #bends = 2
(7, 7), (7, 6), ..., (11, 11)
Interconnection 3: fails.
```

- 每條 Net 必須依序輸出，成功繞線需包含長度、彎折數及路徑，失敗則直接輸出 `Interconnection x: fails.`
- 路徑以 `(x, y)` 依序列出，座標之間以逗號隔開。

## 2. 輸出合法條件（必須同時滿足下列所有條件）

1. **路徑端點正確**：每條 net 路徑的起點、終點須對應 input 指定座標。
2. **路徑連續且單純**：每個相鄰點必須曼哈頓距離為 1，且每個點僅走過一次（不可自交或繞圈）。
3. **不穿越障礙物**：路徑上的所有點不可在任何 blockage 區域內。
4. **不重疊**：所有已繞線成功的 nets 路徑不可互相重疊（每格只被一條線使用）。
5. **範圍合法**：所有座標必須在 1 ~ 寬度、高度範圍內。
6. **output 與 input net 數量一致**，每條 net 皆需依序輸出結果（失敗的以 failed 標示）。
7. **統計資訊需正確**：
  - `#interconnections routed` = 成功路徑數

- `Total interconnection length` = 所有成功路徑長度加總
- `The longest interconnection` = 最長路徑編號及長度
- `Total number of bends` = 所有成功路徑彎折數加總
- 每條 net 的 length 與 bends 需與路徑吻合

---

### 3. 執行資源與限制

- 最大寬/高 (W, H) = 64
- 最大 blockage 數 = 40
- 最大 net 數 = 30
- 執行時間限制：20 分鐘
- 記憶體上限：32 GB
- 時間複雜度: 無限制
- 程式語言: C++ or Python
- 作業系統: Linux

---

### 4. 作業規則與評分標準

#### 1. 測試資料說明

- 共計 10 筆測資，涵蓋基本與邊界情境。
- 每筆測資各佔 10 分，總分 100 分。

#### 2. 編譯與執行規範

- 助教將使用 `make` 指令進行編譯。
  - 考量到本作業難度較高，請同學**自行撰寫 Makefile**。助教批改時將統一使用 `make` 指令進行編譯。**若無法順利編譯，該作業將不予記分。**
  - 請確保執行檔的名稱是 `a.out`。
  - 若使用 Python 撰寫，請確保主程式檔名為 `main.py`。
- 程式輸入輸出需透過**檔案讀寫**，並能以下列指令執行：

```
./a.out [input_file] [output_file]
```

或 (Python 版本)：

```
python main.py [input_file] [output_file]
```

#### 3. 繳交與命名規範

- 請將作業壓縮成 ZIP 格式，檔名須為 `student_id.zip`，例如：`312345678.zip`。
- 所有檔案命名、壓縮格式及內容須完全符合規定，否則將視情節予以扣分。

#### 4. 評分標準

1. 本作業屬於 NP-Hard 問題，並不要求每位同學都能找到最佳解。我們將採用排名式評分制度，鼓勵大家積極優化演算法、追求更佳結果。
2. **不合法的輸出：0 分**
  - 任何不符規範的情形（如路徑不連續、座標超出範圍、格式錯誤、不同 net 有重疊、統計資訊與實際路徑不符等）皆不予計分，所以請務必確保格式、數據、路徑皆正確。
3. **合法的輸出：70% 分數**
  - 只要繳交合法且正確的解，即可取得該筆測資 70% 的分數。
4. **剩餘 30%分數將根據品質排序分配**
  - 根據下列順序進行品質排序：
    1. routed（可繞出的 net 數，越多越好）
    2. total length（總線長，越短越好）
    3. longest length（最長路徑長度，越短越好）
    4. bends（總彎折數，越少越好）
  - 採用標準化排名：根據所有同學在該筆測資的表現排序，第一名獲得30% 滿分，其他同學依序線性遞減至 0%。

---

## 5. evaluator.py 使用方式

### (1) 執行方式

請在 Linux 環境下使用，並將 `evaluator.py`、`input`、`output` 放在同一目錄，終端機執行：

```
evaluator [input_file] [output_file]
```

範例：

```
evaluator sample.in sample.out
```

建議於校內計算伺服器或 Ubuntu 系統上操作

### (2) 會自動檢查項目

- 路徑與統計資料完全正確
- 不穿越障礙物、不重疊、不超出範圍
- 路徑連續性、無自交
- 每個 Net（不論成功或 failed）都正確依序輸出
- 所有統計資訊與實際繞線狀況吻合

### (3) 驗證通過/失敗輸出

- 通過時顯示：`All nets and statistics are routed and reported legally.`
- 不合法時顯示對應錯誤訊息，請依訊息修正
- 額外功能：查詢當前排名分布
  - 當 evaluator 驗證通過後，程式會詢問：「是否查詢當前排名(y/n)?」
  - 若回答 `y`，則會自動顯示目前本測資在所有已繳交結果中的排序情況
  - 僅作為測試使用，不列入最終評分

---

## 6. 常見錯誤提醒

- Net 輸出順序與 input 順序不符
- 統計數值未同步更新（如漏計彎折、最長路徑標錯）
- 失敗 net 未輸出或沒用 failed 標示
- 路徑有跳格、重複經過同一點
- 成功繞線但結果標為 failed，或反之

---

## 7. 補充提醒

- 請確保所有 net 皆依序輸出
- 請確保統計數值與結果一致，否則皆視為不合法。
- 系統會以嚴格的自動化評分流程與標準化排名給分。
- 如需測試資料、評分規則進一步說明，請主動洽詢助教。
- 會陸續上傳更多測資供大家測試程式的正確性與效能，請善加利用。
- 本作業所附的 **evaluator** 僅供測試使用，可能仍有錯誤或與正式評分有出入，如有新版會在課程公告，請留意後續更新。

---

如有任何疑問，請盡速聯繫助教，祝大家 Coding 順利 ✨

TA