

台大電機演算法第一組  
2015/6/30 補充  
有關於計分以及檢察著色正確性  
B00404017 湯智帆

```
color A: -5  
color B: -5  
  
color A: 1  
color B: 0  
  
color A: 1  
color B: 0  
  
color A: 1  
color B: 0  
  
CORRECT COLORING!!!
```

(ps: color 為-5 表示無法著色的 edge!)

我們在每一筆資料都有進行 coloring check 就是檢查相鄰 Edge 兩端的 vertex 應該要有不同色，而 iccad 提供的測資五筆經過我們的著色演算法結果都是 Correct coloring。因此根據官方網頁提供的公式

$$f(x) = \begin{cases} 20, & \text{if coloring graphs for test case } x \text{ are built correctly} \\ 0, & \text{others} \end{cases}$$

$$g(x) = \begin{cases} 10, & \text{if color density windows for test case } x \text{ are shown correctly} \\ 0, & \text{others} \end{cases}$$

$$h(x) = \begin{cases} \sum_{d=1}^{k(x)} \left| \frac{70}{k(x)} - \frac{|color\_A\_density_d - color\_B\_density_d|}{5} \right|, & \text{if coloring for test case } x \text{ is correct} \\ 0, & \text{others} \end{cases}$$

可以開始著手計算得分

第一筆測資得分：

```
the score of test casecase1 is:  
95.1354
```

第二筆測資得分：

```
the score of test casecase2 is:  
90.5821
```

第三筆測資得分：

```
the score of test casecase3 is:
-45.9988
```

第四筆測資得分：

```
the score of test casecase4 is:
89.4279
```

第五筆測資得分：

```
the score of test casecase5 is:
87.8728
```

而根據上述計算平均公式：

$$final\_score = \frac{1}{n} \sum_{x=1}^n score(x)$$

可以進行平均分數計算

$$\frac{95.1354 + 90.5821 + (-45.9988) + 89.4279 + 87.8728}{5} = 63.40$$

以上是我們針對結果的分數計算部分。

很明顯，我們可以發現有一個 outlier，就是 case3，因為 case3 都是單獨一個一個 vertex 的，因此我們的著色演算法沒有考慮到這點，因此造成分數較低，這是我們之後需要思考與改進演算法的部分。

✧ 補充：有關於計分的澄清，因為我們之後才發現其實他有做一個修正，把原本的有外面絕對值得改成沒有絕對值，因此才會有負數的情形

$$h(x) = \begin{cases} \sum_{i=1}^{n(x)} \left( \frac{70}{n(x)} - \frac{|densityAi - densityBi|}{5} \right), & \text{if coloring is correct in case } x \text{ and its value } \geq 0 \\ 0, & \text{others} \end{cases}$$

## 分工情形

湯智帆

- a. main.cpp 檔案讀取部分以及 main.cpp 後面 color checking 部分
- b. getpart.cpp getpart.h
- c. graph.cpp graph.h(主要由助教作業提供，有些部分如 connected component(cc)由湯智帆修改)
- d. Progress presentation
- e. 後續計分程式，如把 density 從 output 取出的 parse.cpp 以及最終計算成績的 score.cpp

徐喆

- a. main.cpp 著色實作，包含 BFS 確定是否 colorable 以及向量著色演算法的實作以及 coloring density 計算
- b. graph.cpp 內的 sortCCByVecLen、CCCompByVecLen 函數
- c. 後續計分結果討論

鍾瑞輝

- a. 提出向量著色演算法以及 coloring density 計算演算法
- b. 期末報告撰寫
- c. 複雜度分析
- d. 軟體 run time 測試

結語：

因為我們這組只有三個人(有一位中途停修)，因此人力分配也較為吃緊，雖然結果差強人意，不過我們仍在這次專題中學習到很多，也有了分工合作的經驗。