# PA2 Result

B03208022 李友岐

1. How to implement the constraint left-edge algorithm

首先運用ifstream讀取檔案，將所有數值都輸入到一個vector aa儲存，並且計算出共有幾個nets。

再看看每個net的Interval為何，

紀錄每個Interval的左端點與右端點，並將這些值儲存到一個二維陣列bb中。

接下來檢查vertical constraint，

運用矩陣來儲存net與net之間

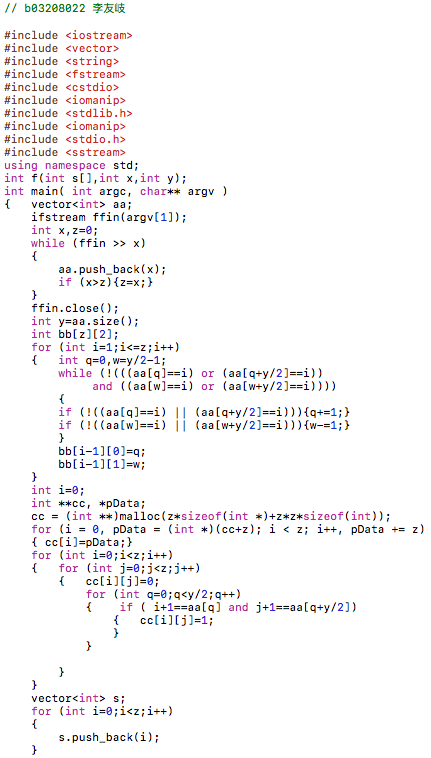
有無vertical constraint。先建立一個動態記憶體配置的二維陣列cc ，

假如i 🡪 j ， cc[i][j]就等於1，

其他部分的值則等於0。

如此即可紀錄net與net間的關係。

再用一個vector s把所有net都儲存起來，s的size就是net的總數。



演算法主要是以兩個while loop達成，其中一個包含在另一個的裡面。我先定義一個function來判斷目前vector s中有無vertically unconstrained 的net，如果有的話d就等於0，反之d則等於1。再建立一個陣列track和變數t，紀錄每個net分配到哪一條track。當d等於0也就是vector s中有vertically unconstrained 的net時，就從其中挑選vertically unconstrained且左端點在最左邊的net，再依序挑選其餘同時符合horizontally unconstrained 和vertically unconstrained 的net 。當net i被選中時，就令track[i]=t，且從s中刪掉net i然後存進另一個vector qw。再讓cc[i][i]=1 ，如此程式就不會挑選已經被挑過的net。當剩下的net都無法同時符合horizontally unconstrained和vertically unconstrained，此輪track的挑選就結束了。最後根據vector qw，假設這輪挑了net i, net j，vector s中還剩net x, net y，就讓cc[i][x], cc[i][y], cc[j][x], cc[j][y]都等於0 。如此即可去除i,j對x,y的vertically constrained ，再根據更新的vector s和更新的二維陣列cc進行下一輪(t+1)挑選。當s中的元素也就是所有的net都被挑選後，s的size會變成0，此時演算法就結束了。所有的net都有分配到ㄧ個track。

最後依序將每個track中有哪些net秀在螢幕上，要注意的是同一個track的net要依左端點的左右順序秀出。

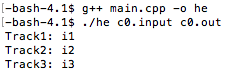
同時也將結果存進一個output file。

我用此function來判斷目前vector s中有無vertically unconstrained 的net。輸入cc到此function中，分別把每一欄內的數值加總。假如net j是vertically unconstrained，那第j欄的數值加總(sum)後必為0。如果偵測到任一欄的sum為0，就回傳0。如果每一欄的sum都不為0，就回傳1。

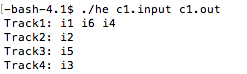


1. The result of the test cases

c0.input



c1.input



c2.input

