* Pseudocode

Algorithm Brute-Force Legalization

curr\_row = 0

for each merged FFi

remove FFs that will be merged from original place

repeat

result = MLL(FFi, tar\_x, tar\_y)

if no exceed boundary

tar\_x = tar\_x + 10;

else

try insert to next row by update curr\_row

update tar\_x and tar\_y

until result = true

tar\_x = FFi->x + FFi->width

end for

Procedure MLL(FFi, tar\_x ,tar\_y)

OK = true

set tFF’s coordinate as (tar\_x, tar\_y)

for each rowi that will be placed by FFi

for each cell ci on rowi

if isOverlap(ci, tFF)

OK = false , and break whole for loop

end for

end for

if OK = true

set FFi’s coordinate as (tar\_x, tar\_y)

insert FFi into corresponding rows

return true

else

return false

* Time complexity analysis

N: number of merged FF

R: number of row height of FFi

C: average number of cells per row

T: average number of iterations required to find a valid placement for a merged FF.

Outer Loop:

1. remove FFi

find: O(log C)

remove: O(C) // vector的插入刪除  
 sort: O(R C log C) = O (C log C) since usually R << C

2. loop for FFi legalization: O(T) \* O(MLL)

MLL:

1. For each cell on row

check isOverlap: O(C)

total cost for R rows: O(R C) = O(C) since usually R << C

1. Inserting FFi into rows

binary search for insertion place: O(log C)

insertion: O (C)

cost per row: O(C + log C) = O(C)

total cost for R rows: O(R C) = O (C) since usually R << C

O(MLL) = O (C)

Total cost for the algorithm = O (N T C)

T 與 no. of site per row 以及no. of row 成正比

worse case of T : O(site num \* row num)

* Special features of your program

1. 檢查FF欲插入的位置是否有Overlap時，不須每次都從FF的min row到max row全部檢查完才有結果，只要找到任一cell與之Overlap就return false代表插入失敗，調整座標後進行下一次嘗試。
2. 每次插入FF後對對應的row cells進行x座標排序時，不會每次都push back再重新排序O(nlogn)，而是使用lower\_bound 二分搜索算法，用來在有序範圍中找到不小於（即大於等於）目標值的第一個位置，然後再insert O(n)。時間複雜度由O(nlogn) -> O (n)
3. 若目標位置會發生overlap，小幅度右移或從下一行開始，若成功插入則下一個FF插入的嘗試位置從上一個插入的FF的右邊邊緣開始，而不用一直小幅度右移嘗試。若該row已遍歷到底，下一個FF也會從下一個row開始嘗試，而非從頭來過。

* Feedback

一開始的attribute of placement row較複雜，又沒有對應條件的testcase，難以下手。

* Conclusion

使用稍微優化過的Brute-Force Approach，FFi從min row開始到max row由左而右嘗試，下一個FFj從FFi的位置開始嘗試，若超出max row則再回到min row。

若沒有做cell movement，用local legalization可能慢又結果不好，不如global legalization。