- Pseudocode

```
Algorithm Brute-Force Legalization
curr_row = 0
for each merged $FF_i$
    remove FFs that will be merged from original place
    repeat
        result = MLL($FF_i$, 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 = $FF_i$->x + $FF_i$->width
end for

Procedure MLL($FF_i$, tar_x ,tar_y)
OK = true
set tFF's coordinate as (tar_x, tar_y)
for each $row_i$ that will be placed by $FF_i$
    for each cell $c_i$ on $row_i$
        if isOverlap($c_i$, tFF)
            OK = false , and break whole for loop
    end for
end for

if OK = true
    set $FF_i$'s coordinate as (tar_x, tar_y)
    insert $FF_i$ into corresponding rows
    return true
else
    return false
```

- Time complexity analysis

N: number of merged FF

R: number of row height of FF_i

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 FF_i
    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 FF_i 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
2. Inserting FF_i 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，FF_i 從 min row 開始到 max row 由左而右嘗試，下一個 FF_j 從 FF_i 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。