

1. 這題要算出 N 位客人得消費總和，每人的消費是在 M 列中選出座位最多的一列並付該列剩餘座位數量的前，且必須用 `max-heap` 的結構來實作。
在程式中先將所輸入的每列位置數量放到 `max-heap` 中，因為客人選的是最多座位的列，所以 `max-heap` 的 `root` 就是花費，將花費累加到 `money` 這個 `array`，接著就將 `root` 的值減一再使用 `Heapify` 這個 `function` 來確保是 `max-heap`，最後 `print` 出 `money` 內的結果。
2. 這題要寫出 `collapsingFind` 和 `heightUnion` 這兩個 `function`。在課本的 `collapsingFind` 會有多餘的 `link reset`，所以在 `function` 中加上 `lead==root` 時就 `break` 的判斷來避免多餘的 `reset`。`heightUnion` 不同於 `weightedUnion` 是根據 `node` 數量來 `Union`，`heightUnion` 是根據 `tree` 的 `height` 來進行 `Union`，這種方法在進行 `collapsingFind` 時能加快搜尋較底下的 `node` 的時間，避免有些 `level` 很大的 `node` 存在。