B06705011 鄭宇翔 HW6

因為天際線的變化只會在每棟大樓的左界或右界發生，所以我們可以只考慮在這 2*n 個點上會發生的事情。

天際線取決於該點最高的大樓高度，所以我們可以考慮用 priority queue 來存這些高度，遇到左界則將該大樓的高度放進 priority queue，遇到右界將該高度取出，如此 priority queue 的 top 可以保持是該點的天際線高度。

雖然演算法可以求出天際線，但是 priority queue 不支援將一個特定的 value 取出，所以在遇到右界的時候，我們不直接將該高度取出，而是用一個 unordered map 紀錄該高度要被取出的次數，每遇到一次該高度的右界就加一次，要輸出一段區間的天際線的時候，先檢查 priority queue 的 top 的值要被刪除幾次，持續 pop out 直到 unordered map 對應的 value 為 0 為止，代表當前的最大高度還沒到右界，所以將其輸出為天際線。

具體的做法是：

兩個 pair 陣列，left_node 存左座標及高度，right_node 存右座標及高度

while 迴圈跑這兩個 pair 陣列，先取出座標比較小的 pair，若左右的座標一樣擇一率先取左邊的。

宣告一個 priority queue max_height，先把 0 放進去當初始值，在宣告一個 unordered_map to_deleted，再宣告一個 vector ans 做為答案的容器

根據取出的 pair 分兩種情況，若是左界，則將 height 加入 max_height，若是右界，將 to_deleted 中 key 為高度的 value 加一，然後 while 迴圈檢查 max_height，若 to_deleted[max_height.top()]>0，代表這個高度早該被取出，所以 max_height.pop()，直到 to_deleted[max_height.top()]=0 為止

previous 為在讀進此節點之前的天際線高度，初始值為 0，current=max_height.top()為現在天際線高度，若 current==previous，代表天際間沒變，這個高度的區間還不用輸出，但是如果 current!=previous，就必須輸出這段區間，將該點座標及高度放進 ans 中，然後令 previous=current

在 left_node 及 right_node 都檢查完以後，將 ans 輸出便是答案。