**Part I: Sort Algorithm**

**1. Introduction**

以下將介紹兩種熱門的sorting algorithm：merge sort與quick sort，這兩者在所有排序演算法中，擁有優秀的平均時間複雜度(average time complexity)，為O(nlogn)，然而這兩者各有優缺，下面進行詳細介紹。

**2. Implement Details**

2- 1. Quick Sort

圖1為Quick sort的程式碼，此演算法採用了分治法(divide-and-conquer algorithm)，會將傳入的array拆成兩個subarray，再分別對兩個subarray進行遞迴排序；Quick sort的function中可分成四大部分：遞迴出口、選擇基準(pivot)、根據基準劃分兩個subarray、將subarray再傳入Quick sort的function。

傳入值上，被排序的array以pass-by-reference的方式傳入，兩個int參數left, right分別代表被排序的array index範圍，最小值是left，最大值是right，圖像上可代表被排序array的最左端與最右端，因此當left >= right時，代表排序範圍只有1個或沒有element，即為遞迴出口條件；選擇pivot上，我是以排序範圍內index中間值所對應的element作為pivot；partition上，會將排序範圍內element >= pivot放在array右半部，element <= pivot放在左半部，操作上我使用了左指針i與右指針j，分別從排序範圍的最左與最右端往中心掃描，在兩指針相遇之前，左指針遇到array[i] >= pivot則停下來，右指針遇到array[j] <= pivot則停下來，接著兩指針所對應的element

一張含有 桌 的圖片

自動產生的描述

圖1. Quick sort程式碼

進行交換，交換完後指針繼續前進，重複以上動作直到兩指針超過彼此；partition結束後，(left, j)和(i, right)分別代表兩個subarray的index範圍，再將這兩個subarray傳入quick sort的function。

2-2. Merge Sort

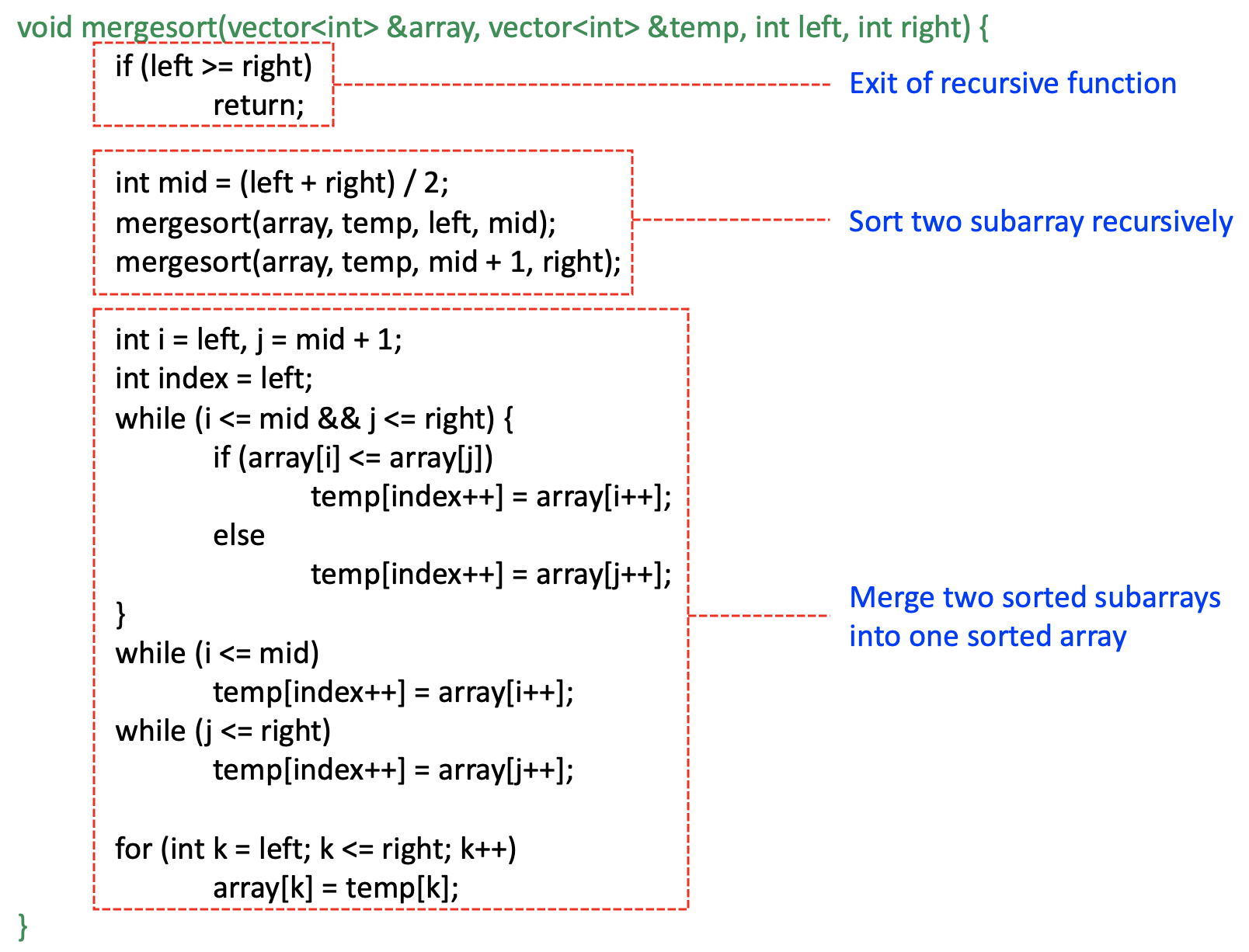


圖2. Merge sort程式碼

3. Results

4. Discussion

5. Conclusion

**Part II: Minimum Spanning Tree Algorithm**

1. Introduction

2. Implement Details

2-1. Prim Algorithm

2-2. Kruskal Algorithm

3. Results

4. Discussion

5. Conclusion

**Part III: Shortest Path Algorithm**

1. Introduction

2. Implement Details

2-1. Dijkstra’s Algorithm

2-2. Bellman-Ford Algorithm

3. Results

4. Discussion

5. Conclusion