作業九

求2-way merging problem的時間複雜度。

**Algorithm 2-way-merge**

Input: k sorted lists

Output: an optimal 2-way merge tree

Method:

Step 1: Generate k trees, each tree has a node with weight (the length of the list)

Step 2: Choose two trees and with minimum weights;

Step 3: Create a new tree T whose root has and as its subtree, and the weight of T is the sum of weights of and ;

Step 4: Replace and y T;

Step 5: If there is only one tree left then stop; else goto Step 2

**Time complexity for generating an optimal 2-way merge tree: O(n log n)**

作業十

求minimal cycle basis problem的時間複雜度。

**Algorithm:**

Step 1: Find all pairs shortest paths.

Step 2: For each vertex v V and for each edge (x, y) E, the cycle consists of shortest path (v, x), edge (x, y) and shortest path (y, v) is a candidate. Compute the weight of each candidate.

Step 3: Sort all candidate cycles. (by weight)

Step 4: Add the candidate cycles to the minimal cycle basis one by one.

**Time complexity = O()**

|V| = n, |E| = m (m = O() in the worst case).

Step 1: O()

n Dijkstra’s algorithm (=O())

Floyd algorithm O()

Step 2: O(), find cycles and their weights

Step 3: O(mn log(mn)) sorting mn cycles

Step 4: O(msmn), s = m-n+1 (the size of basis) = O()