

2003

1.(a)complete binary trees: complete binary tree 中所有的樹葉節點都位於相鄰的階層上

Or 一個有 n 個節點且深度為 k 的二元樹是 complete binary tree 若且唯若它的節點和一個深度為 k 的 Full binary tree 中，從編號 1 到 n 的節點一致。

(b)min heaps: min heap 是一種每個節點的鍵值都不比它子節點(如果存在的話)大的完整二元樹。

(c)spanning tree: 當 G 為連通圖時，任何一棵只包含 G 裡的邊以及 G 裡的所有頂點的樹稱為生成樹。

(d)external sorting: 串列太大時所使用的排序法

(e)a connected component of an undirected graph: 一個無向圖的連通元件 H 指的是這個無向圖中的極大(maximal，是指在無向圖 G 裡不存在一個連通子圖包含 H 且不等於 H)連通子圖。

(f)dynamic hashing: Can accommodate dynamically increasing and decreasing file size without penalty

(g)level order traversal: 使用 stack 且依照 Full binary tree 的節點編號順序來拜訪各個節點的走訪方式

(h)AOV networks: 一個有向圖 G 中，其頂點代表活動或任務而邊代表各個任務間的先後關係，稱此...

2004

1.(a)internal sort:串列夠小到可以整個放到主記憶體中來進行排序時所用的方法

(b)biconnected graph: biconnected graph 沒有接合點(articulation point)的圖

(c)uniform hash function: The hash table is fixed-sized. For a randomly chosen key, k , the probability that $h(k) = i$ is $1/b$ for all buckets i .

(d)min-max heap: min(max) heap 是一種每個節點的鍵值都不比它子節點(如果存在的話)大(小)的 complete binary tree。

(e)topological order: 拓撲順序是一個圖中頂點的線性順序，它滿足對於任兩點 i 與 j ，如果 i 是 j 的先行點，則在此線性順序中 i 排在 j 之前。

(f)left out of order(LOO): 目前輸入的值是比原有陣列各個的值還小

2005

1.(a)simple path: 如果在一條路徑上除了第一點與最後一點外，其餘的頂點都不相同，稱此路徑為...

(b)articulation points: G 是一個無向的連通圖，當 G 中的一個頂點 v 是接合點若且唯若刪掉 v 以及所有接附在 v 的邊時將產生至少有兩個連通元件的圖 G'

(c)static hashing: The hash table is fixed-sized.

(d)AOE networks: 在計畫中要執行的任務用有向邊表示，而網路上的頂點代表事件。當事件發生意味著某項活動已經完成。而從某個頂點的有向邊所代表的活動，只有在該頂點所代表的事件發生後才可開始執行。

(e)spanning tree: 同上

(f)connected component: A maximal connected subgraph

2007

1. (b)AOE networks: 同上

(c)static hashing: 同上

(d)complete graph: 一個有 $n(n-1)/2$ 個邊的 n 個頂點的無向圖，稱為...

(e)uniform hash function:同上

(f)spanning trees: 同上

(g)critical activity: 所有最早開始時間等於最晚時間($e(i) = l(i)$)的活動稱為...

(h)internal sort: 同上

(i)left out order(LOO): 同上

(j)connected component: 同上

2. 自己翻課本

3.(a) 例如一棵有三節點的樹，root 為 10，左小孩跟右小孩都是 5，那麼第一次的 pop 後會抓右小孩遞補，那麼第二次的 pop 就會造成，原本在後面的 5 跑到前面(第二個)

(b)因為是最小連通圖，又要包含所有 vertex 而 vertex 有 n 個，所以 edge 有 $n-1$ 個

(c)每次選 pivot 的時候，都使得分邊的狀況是歪斜於一邊，所以需要選 n 次 pivot，每次的 pivot 需要 reorder 一次輸入資料，故需 $O(n^2)$

4.(a)F，excluding \rightarrow including

(b)F，自己翻課本 p.6-59 下面

(c)超出範圍

(d)F，要除以二

(e)F，因為 dfs 或 bfs 所經過的路徑並不一定會經過 minimal cost 的邊

5.(a)sum of row[i] or col[i]

(b)(1)計算簡單，減少碰撞 (2)計算雜湊函數與搜尋一個桶所需要的時間

6.自己翻課本

7.(a)從 $dist^1$ 開始，先將可以到達的 vertex 記錄於 array 中，無法到達者設為無限大，再來以前述資料尋

找 $dist^2$ ，以此類推 $dist^k[u] = \min\{dist^{k-1}[u], \min\{dist^{k-1}[i] + length[i][u]\}\}$ ，直到 $k = n-1$ 為止 (n 為 vertex 數)

(b) $\square O(n^3)$ when adjacency matrices are used

$\square O(ne)$ when adjacency lists are used

\square Complexity reduction

\square Terminate the for-loop either after $n-1$ iterations or after the first iteration in which no $dist$ values are changed

\square Maintain a queue of vertices i whose $dist$ value changed on the previous iteration of the for-loop.

\square The only values for i that need to be considered during the next iteration

8.(a)超出範圍

(b) Decompose a problem into subproblems and each subproblem will be solved by the same approach

recursively

9.(a) e(i) 6,4,5,7,7,16,14,18

(b) $l(1)=6$

$l(2)=6$

$l(3)=8$

$l(4)=7$

$l(5)=10$

$l(6)=16$

$l(7)=14$

$l(8)=18$