(a) AVL trees

AVL tree is height balanced tree. A height-balances tree a nonempty binary tree with TL and TR as its left and right subtrees respectively, then T is height-balanced iff(1)TL and TR are height-balanced tree (2)
|hL-hR|<=1 where hLhR are the heights of TL and TR, respectively
沒交不會寫

(b) AOE networks

AOE network is a directed graph G in which the vertices represent event and the edges represent the task to be performed on a project
AOE network 是一個有向圖 G, G 的一個 vertices 代表一個事件,一個邊代表一個活動, 這個邊跟活動能完成一個 project

(c) Static hashing(ASK)

A type of hash which the hash table is fixed-sized
雜湊是一種搜尋方法 可以盡量減少搜尋範圍到只剩下一個 換句話說 雜湊技術如同陣列一般 可以直接存取檔案結構的紀錄資料,而靜態雜湊是雜湊的一種,他的雜湊表是 fixed-sized

(d) Complete graph(ASK)

一個完整的圖,並非生成樹,它包含了一個 G 圖裡最大數量的邊和 G 裡所有的頂點

(e) Uniform hash function

For a randomly chosen key, k, the probability that h(k) = i to be 1/b for all buckets i.
假如 k 是從鍵值空間裡隨機挑選出來的鍵值,如果對於所有桶 i 來說,h(k) = i 的機率都是等於 1/b 代表一個隨機的鍵值雜湊到任何一個桶的機率都相等,稱為均勻雜湊函數

(f) Spanning trees

a spanning tree is any tree that consists solely of edges in G and that includes all the vertices in G..
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

(g) Critical activity

在 AOE 中完成整個計畫的最少時間是從開始頂點到完成頂點的最長路徑,此路徑稱臨界路徑,而在此路徑上的所有活動 a(i)都稱為 Critical activity

(h) Internal sort

當 sort 資料量大小未超過主記憶體負荷時(及 sort 所需要的資料量皆可存放在主記憶體中執行 sort)，即為 internal sort

(i) Left out of order(LOO)

Record Ri is left out of order(LOO) iff Ri<max{Rj}1<=j<i

紀錄 Ri 被稱為左失序若且唯若 Ri<max{Rj}

(1<=j<i)max 的條件

(j) Connected components

Connected component is the maximal connected subgraph of a graph

連通元件是在一個圖中的最大子樹圖

2.(課本 288 的圖)

(a)

Vertex	0	1	2	3	4	5	6	7	8	9
Dfn	4	3	2	0	1	5	6	7	9	8
Low	4	0	0	0	0	5	5	5	9	8

(b)

Articulation points: 1 , 3 , 5 , 7

Biconnected component: (課本 287.圖 6.19(b))畫出那六個元件

3

(a) (2%) Heap sort is not stable. Give an example of an input list in which the order of records with equal keys is not preserved.

(b) (3%) Show that the worst case time complexity of quick sort is O( ), while the best case time complexity is O(*n* log *n*).

(c) (2%) Explain why a spanning tree contains exactly n-1 edges.

(a)(26,5,12,10,26')->(5,10,12,26',26)

(b) Worst case: T(n)=T(n-1)+cn

(c)Each node in the tree has a edge pointing to its parent instead of the root.

(簡單來說 spanning tree 須包含 G 裡所有的頂點,而且要把它們全部連起來,也就是不能有落單的頂點,所以是 n-1 個邊)

4 對或錯

T (a)Given a graph and a vertex, DFS and BFS may obtain distinct results excluding the sequence.

F(b) If an AOV network represents a feasible project, it means that there is a unique topological order for the network.

T(c)大概是說所有溢位桶裡的鍵值都不同的最小 u 值,如果這個最小 u 值比目錄深度大,這使我們必須加大目錄的大小,但不用增加桶數

F(d) Let be the degree of vertex in a graph G with $|V| = n$ and $|E| = e$, then $e = \sum_{i=}^{n-}$

F(e) The path from vertex A to vertex B on a minimal cost spanning tree of an undirected graph G is a shortest path from A to B.

5.

(a) With adjacency matrices representation, how to determine the degree of a vertex in an undirected graph?

(b) What are the two principle for choosing a hashing function?後面問的是有兩件事可以決定 hashing 中 insert,delete,search 的時間

(a).舉例:

	0	1	2	3	
0	0	0	1	1	->0 的 degree = 2
1	0	0	0	1	->1 的 degree = 1
2	1	0	0	1	->2 的 degree = 2
3	1	1	1	0	->3 的 degree = 3

(b).Make no collision and overflow. The function will distribute the data evenly.(課本寫的不知道哪個對 1.容易計算且將碰撞次數降到最低 2.在隨機輸入的情況下不會產生雜湊表的偏頗使用情況)

(計算 hash 跟 search bucket)

6.

Consider the 2-way merge on disk. Assume there are 12000 records in disk to be sorted using a computer with an internal memory capable of sorting at most 1000 records. Also assume that disk I/O is with block length of 250 records. Let be the I/O time, including maximum seek time, maximum latency time, and transmission time, for a block of records. What is the total I/O time for the external sorting?

1.

read 48 blocks of input, $48t_{IO}$ $96t_{IO}+12t_{IS}$

internally sort, $12t_{IS}$

write 48 blocks, $48t_{IO}$

2.

Merge runs 1 to 12 in pairs $96t_{IO}+12000Tm$

3.

Merge two runs of 2000 records each $96t_{IO} + 12000Tm$

48blocks *3 次

4.

Merge two runs of 4000 records with 48 blocks $64t_{IO} + 8000Tm$

5.

8000 跟 4000 合併 $96t_{IO} + 12000Tm$

total $448t_{IO} + 44000\ Tm + 12t_{IS}$

7.第一小題要用 Bellman&ford 去設計類似課本圖 6.31 的問題

```
for(int i =0;i<n;i++)
      dist[i] = length[v][i];    //初始化
for(int k = 2;k<=n-1;k++)
    for(每個 u 滿足 u!=v 且 u 至少有一個進到他的邊)
       for(圖上的邊<i,u>)
            if(dist[u] > dist[i] + length[i][u])
               dist[u] = dist[i] + length[i][u];
```

(抄課本的)

第二小題要問用不同的呈現方法時, Bellman&ford 會有不一樣的複雜度

然後問說有什麼方法可以減少 Bellman&ford 的複雜度

如果用相鄰矩陣,複雜度是 O(n^3)

相鄰串列,複雜度是 O(ne)

他課本寫說用動態規劃法,因為這樣他的常數因數比較小

8.

(a)(10%) Draw the tree configurations for the rotations *RL* of AVL trees. [Hint: similar to the three abstract cases of LR listed in the textbook.]

(b)(5%) Explain how to apply dynamic programming method to solve the all pairs shortest paths problem. **(p.308)**

(a)聽說沒交我也不會

(B)for(k=1;k<=vertexnum;k++)

```
For(i=1;i<=vertexnum;i++)
For(j=1;j<=vertexnum;j++)
If(adj[i][k] && adj[k][j])d[i][j]=men{d[i][j],d[i][k]+d[k][j]};
```

9

反正就是課本圖 6.40

答案是圖 6.42

臨界路徑:0->1->4->6->8

0->1->4->7->8(注意路徑的數字代表的是 vertex)