《前言》

1. 依照期末考公告，考試範圍為投影片第 5 章 Page 16 到第 8 章 Page 11，但實際上課進度為 Page 17。
2. 新年快樂 ～

1. Explain the following or terms comparisons:
   (a) Max / Min trees VS. Max / Min heaps

   < 英文版 >

   Max(Min) trees:

   A max (min) tree is a tree in which the key value in each node is no smaller (larger) than the key values in its children (if any).

   Max(Min) heaps:

   A max (min) heap is a complete binary tree that is also a max (min) tree.

   < 中文版 >

   Max(Min) trees:

   每個節點的 Key 都不比子節點小(大)的二元樹

   Max(Min) heaps:

   Max(Min) trees 滿足 Complete binary tree 之條件

   (b) Binary search trees

   < 英文版 >

   A binary search tree is a binary tree. If it is not empty it satisfies the following properties:

   - Every element has a key, and no two elements have the same key, i.e., the keys are unique.
   - The keys in a nonempty left subtree must be smaller than the key in the root of the subtree.
   - The keys in a nonempty right subtree must be larger than the key in the root of the subtree.
   - The left and right subtrees are also binary search trees.

   < 中文版 >

   一棵二元樹，若不為空則有以下性質：

   - 每個元素都有一個 Key 且不存在兩個元素有相同的 Key
   - 左子樹裡所有的 Key 皆小於當前節點內儲存的 Key
   - 右子樹裡所有的 Key 皆大於當前節點內儲存的 Key
   - 左右子樹都是 Binary search tree

   (c) Binary Tree traversal

   < 英文版 >

   Visiting each node in the tree exactly once.

   Notations:

   - L -- Moving left
   - V – Visiting the node
   - L -- Moving right

   Three possible traversals if we traverse left before right:

   - LVR (inorder)
   - LRV (postorder)
   - VLR (preorder)

< 中文版 >

拜訪樹中每一個 Node 一次而已，且若假設 L、R、V 分別代表「往左走」、「往右走」、「拜訪節點」，加上傳統先左後右的走法，就會有三種走法：

- LVR (中序)：可產生出 infix 的表達式。
- LRV (後序)：可產生出 postfix 的表達式。
- VLR (前序)：可產生出 prefix 的表達式。

(d) AVL Trees (超出範圍，講義未出現)

< 中文版 >

AVL tree 是一種自平衡二元搜尋樹，特性是所有節點的左子樹與右子樹高度差不超過 1，在搜尋、插入以及刪除時可以加快速度。

(e) Spanning Tree

< 英文版 >

Any tree that consists solely of edges in graph G and that includes all the vertices in G.

< 中文版 >

任何一棵只包含 G 裡的邊以及 G 裡所有頂點的樹，稱之為生成樹。

(f) Connected components

< 英文版 >

Two vertices u and v are connected in an undirected graph G if there is a path in G from u to v.

A maximal connected subgraph.

< 中文版 >

在無向圖 G 中，若頂點 u 與 v 間存在一條路徑，則稱之為相連。

連通元件則是圖 G 中極大的相連子圖。

(g) Complete Graph

< 英文版 >

A graph that has the maximum number of edges.

The maximum number of edges for graph with $|V| = n$:

Undirected = $\frac{n(n-1)}{2}$, Directed = $n(n-1)$

< 中文版 >

具有最大邊數的圖。倘若有 n 個頂點且為無向圖，則最大邊數是 $\frac{n(n-1)}{2}$，反之若為有向圖，則最大邊數為 $n(n-1)$。

(h) Articulation Points

< 英文版 >

A vertex v such that the deletion of v, together with all edges

incident on v, produces a graph G' having at least two connected components.

< 中文版 >

刪除頂點 V 以及其相連的邊，若會產生至少有兩個連通元件的圖，則頂點 V 稱為接合點。

(i) Simple Path

< 英文版 >

A path in which all vertices, except possibly the first and the last, are distinct.

< 中文版 >

除了起點與終點以外，沒有重複頂點的路徑。

(j) AOE Networks

< 英文版 >

The directed edges represent tasks or activities to be performed on a project.
The vertices represent events which signal the completion of certain activities.
The activities represented by edges leaving a vertex cannot be started until the event at that vertex has occurred.

< 中文版 >

AOE 網路圖是一個有向圖 G，在 G 中的頂點表示事件，邊表示活動，而這些事件與活動能夠代表完成一個專案。

(k) AOV Networks

< 英文版 >

An AOV network is a directed graph G in which the vertices represent activities and the edges represent precedence relations between activities.

< 中文版 >

AOV 網路圖是一個有向圖 G，在 G 中的頂點表示活動，而邊則表示這些活動間的優先關係。

(l) Predecessors / Successors

< 英文版 >

Predecessors: Vertex i in an AOV network G is a predecessor of vertex j iff there is a directed path from i to j. Vertex i is an immediate predecessor of vertex j iff <i, j> is an edge in G.

Successors: If i is a predecessor of j, then j is a successor of i. If i is an immediate predecessor of j, then j is an immediate successor of i.

< 中文版 >

Predecessors: 當存在一條從 u 至 v 的有向路徑時，u 稱為 v 的先行者。此外，若存在一條邊 <u, v> 的話，則 u 稱為 v 的直接先行者。

Successors: 若 u 為 v 的先行者，則 v 是 u 的繼承者。此外，若 u 為 v 的直接先行者，則 v 是 u 的直接繼承者。

(m) Transitive / Irreflexive

< 英文版 >

A relation · is transitive iff for all triples i, j, k, $i \cdot j$ and $j \cdot k \rightarrow i \cdot k$.

A relation · is irreflexive on a set S if $i \cdot i$ is false $\forall i \in S$.

< 中文版 >

關係 $i \cdot j$ 是指 i 必須在 j 之前完成。

若 $i \cdot j$ 且 $j \cdot k$ 則 $i \cdot k$，稱之為遞移性。

對於集合中沒有一個元素會滿足 $i \cdot i$，稱之為漫反射性。故 DAG 具有漫反射性。

(n) Topological order

< 英文版 >

A topological order is a linear ordering of the vertices of a graph such that, for any two vertices, i, j, if i is a predecessor of j in the network then i precedes j in the linear ordering.

< 中文版 >

拓撲排序使得如果存在一條從頂點 i 到頂點 j 的路徑，那麼在排序中 j 出現在 i 的後面。

(o) Critical Path

< 英文版 >

A critical path is a path that has the longest length from the start vertex to the finish vertex.

< 中文版 >

從開始頂點到完成頂點的最長路徑，此路徑稱為臨界路徑。

(p) Critical Activity

< 英文課本版 >

An activity for which e(i) = l(i).

< 中文版 >

在臨界路徑上的所有活動都稱為臨界活動。簡言之，對於一活動 i，其最早活動時間與最晚活動時間相等。

(q) Internal / External Sort

< 英文版 >

- Internal Sort: The list is small enough to sort entirely in main memory.
- External Sort: When there is too much information to fit into main memory. The file must be brought into the main memory in pieces until the entire file is sorted.

< 中文版 >

內部排序：資料量大小未超過主記憶體負荷之時，可將資料儲存於記憶體中執行排序。

外部排序：當資料量過大，無法完全載入至主記憶體之時，則須將資料分批載入，直到整個資料完成排序為止。

(r) Left Out of Order (LOO)

< 英文版 >

R_i is LOO iff $R_i < max_{1 \le j < i}\{R_j\}$.

< 中文版 >

若紀錄 R_i 滿足 $R_i < max_{1 \le j < i}\{R_j\}$ 之條件，則稱為左失序。

(s) Stable Sorting Method

< 英文版 >

A sorting method is stable if the generated permutation σ_s is unique and has the following properties.

- $K_{\sigma_s(i)} \le K_{\sigma_s(i+1)}\ for\ 0 < i \le n-1$
- $If\ i < j\ and\ K_i == K_j\ in\ the\ input\ list, then\ R_i\ precedes\ R_j\ in\ the\ sorted\ list$

< 中文版 >

假設排序演算法所生成的排序 σ_s 滿足以下條件則為穩定的排序法：

- $K_{\sigma_s(i)} \le K_{\sigma_s(i+1)}\ for\ 0 < i \le n-1$
- 輸入序列中若有紀錄滿足 $K_i == K_j$ 且 $i < j$，則在排序後序列中 R_i 將在 R_j 之前。

(t) Static Hashing

< 英文原版 >

A type of hash which the hash table is fixed-sized

< 中文原版 >

靜態雜湊是雜湊的一種，其雜湊表是固定大小的。

< 中文版 >

雜湊是將資料代入事先設計好的雜湊函數，求得此資料所對應的主桶，使得搜尋範圍減少，進而提升相關操作的效率。靜態雜湊也是雜湊的一種，但雜湊表是固定大小。

(u) Uniform Hash Function

< 英文原版 >

For a randomly chosen key, k, the probability that h(k) = i is $\frac{1}{b}$ for all buckets i.

< 中文原版 >

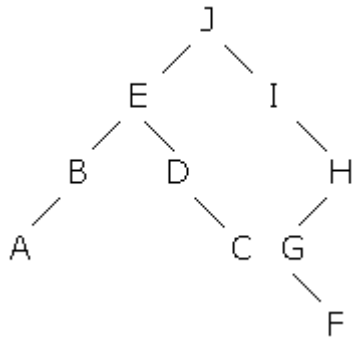
假如 k 是從鍵值空間裡隨機挑選出來的鍵值，如果對於所有桶 i 來說，h(k) = i 的機率都是等於 $\frac{1}{b}$。代表一個隨機的鍵值雜湊到任何一個桶的機率都相等，稱為均勻雜湊函數。

< 中文版 >

若對於鍵值空間中的任一鍵值，經雜湊函數映射到地址集合中任何一個地址的機率是相等的。

2. Given an inorder sequence ABEDCJIGFH and a postorder sequence ABCDEFGHIJ, can you derive a unique binary tree? If yes, draw a binary tree; or you have to give two distinct binary trees which can generate above sequences.

   < Sol >



3. Tree operations

   (a) Describe how to delete an element from a binary search tree. Calculate the time complexity of the deletion operation.

   < Sol >

   若欲刪除的節點為葉子，則直接刪除。反之，則尋找左子樹最大者(Predecessor)或右子樹最小者(Successor)，將其值複製過來後刪除。

   最糟是從最上層找到最底層，所以時間複雜度為 $O(log_2 n)$。

   (b) Describe how to insert an element into a min heap. Calculate the time complexity of the insertion operation.

   < Sol >

   插入節點之時，先假設在 Heap 的最後一個，然後與父節點做比較，如果比較小就往父節點移動。

   最糟是從最底層移動至最上層，所以時間複雜度為 $O(log_2 n)$

4. We can use the tree structure for set representation. In this application, union and find are the minimal operations: the formal is for disjoint set union and the latter is for finding the set containing some specified element. How can we speed up the two operations? (Hint: What rules?)

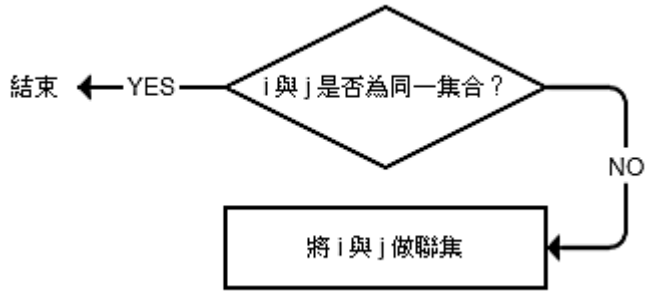
   < Sol >

   Union 操作可透過 Weighting Rule 將節點個數少的 Set 指向節點個數多的 Set，減少聯集後樹的高度。

   Find 操作可透過 Collapsing Rule，每碰到一個節點就把它的 Parent 指向根節點，如此一來便能節省下次進行 Find 的時間。

5. Solving the equivalence classes problem is an application of binary search trees. Explain how to process an equivalence pair, i≡j.

< Sol >



6. Graph Theorem

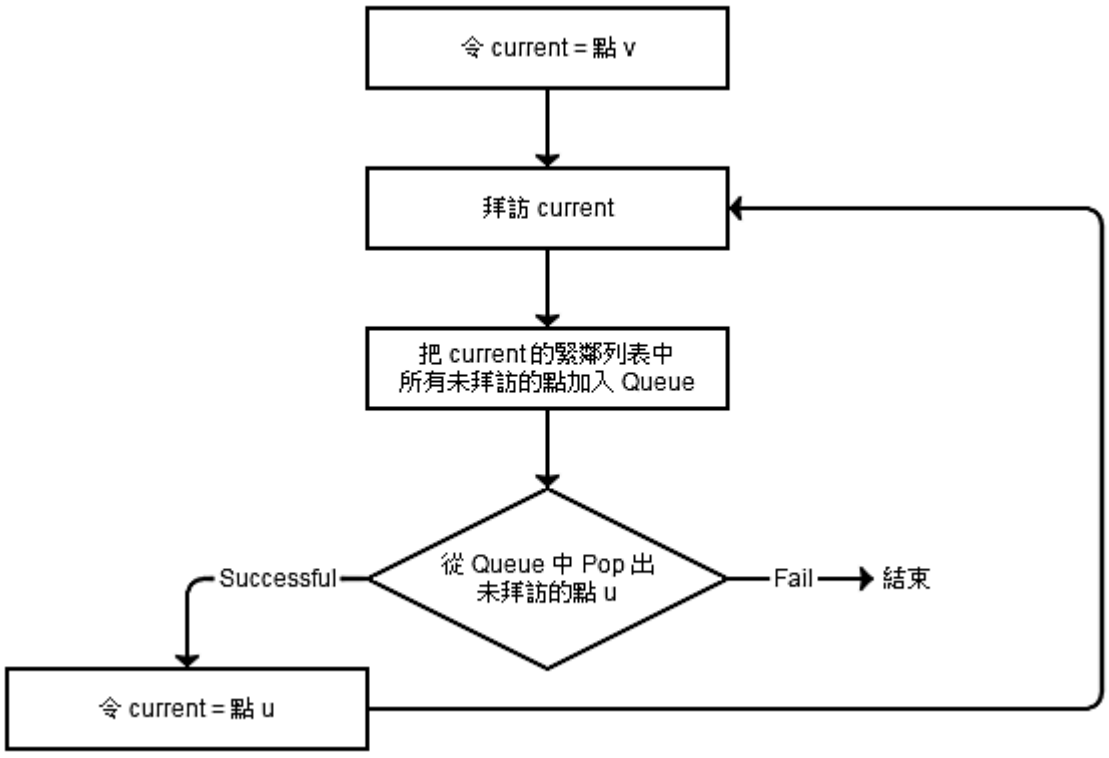
(a) What adjacency matrices representation, how to determine the degree of a vertex i in an undirected graph?

< Sol >

假設緊鄰矩陣為 a，則點 i 的分歧度為 $\sum_{j=0}^{n} a[i][j]$。

(b) Let the start vertex be v and suppose that we use adjacency lists as the graph representation. Describe the breadth first search operation and the required time and complexity.

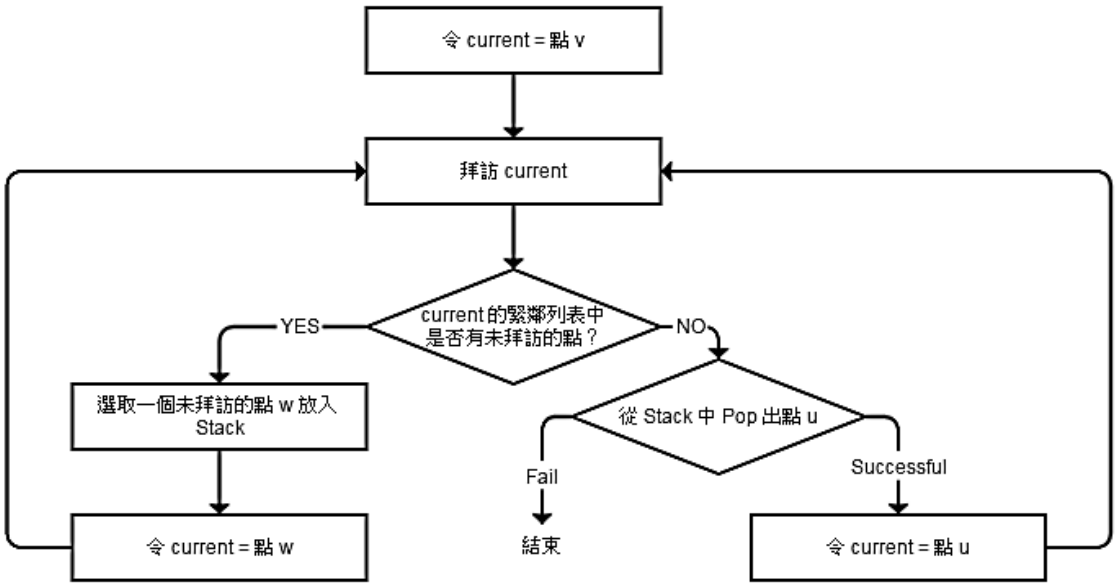
< Sol >



時間複雜度為 $O(V+E)$，其中 E 為圖中邊數。

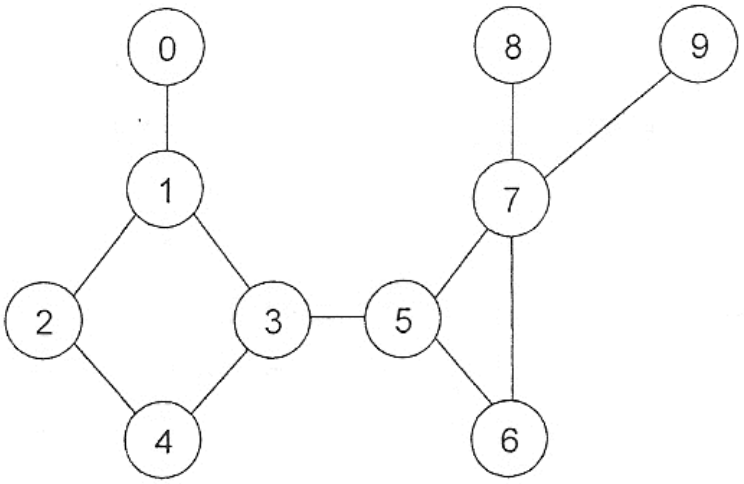
(c) Let the start vertex be v and suppose that we use adjacency lists as the graph representation. Describe the depth first search operation and the required time and complexity.

< Sol >



時間複雜度為 $O(V+E)$，其中 E 為圖中邊數。

7. Depth First Number and Low values



(a) Give the depth first numbers and low values of all nodes in the above graph.

< Sol >

從頂點 3 開始，則表格如下，詳細過程請參考作業 6。

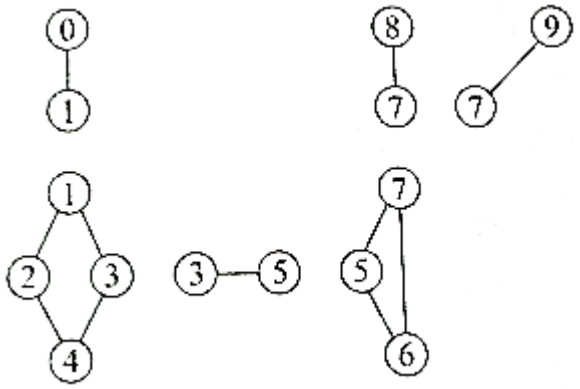
Vertex	0	1	2	3	4	5	6	7	8	9
dfn	4	3	2	0	1	5	6	7	9	8
low	4	0	0	0	0	5	5	5	9	8

(b) List all bi-connected components and articulation points.

< Sol >

Articulation Points: V1, V3, V5, V7

Bi-connected components:



8. According to the definitions of depth first number(dfn) and low value, describe the sufficient and necessary condition of for a vertex u to be an articulation point.

< Sol >

若點 u 為接合點，則符合以下任一條件：

- 點 u 為生成樹的樹根且有兩個以上的兒子
- 點 u 並非樹根，但點 u 存在一個兒子 w，使得 $low[w] \geq dfn[u]$

9. Describe the following algorithms, which construct minimal spanning tree of a given graph.

(a) Prim

< Sol >

- 一開始任選圖中一點開始。
- 每次選取 Cost 最小的邊 (u, v)，使得 u、v 其中一點已包含於最小生成樹 T 中，且 $\{T \cup (u, v)\}$ 亦為一棵樹。

(b) Kruskal

< Sol >

- 將所有邊遞增排序。
- 每次選取 Cost 最小的邊，倘若此邊不會形成環，則將此邊加入最小生成樹 T 中。

(c) Sollin

< Sol >

- 假設一開始各頂點為獨立的 MST。
- 在同一階段中，選取各 MST 的 Cost 最小聯外邊，倘若此邊不會形成環，則將此邊加入生成樹 T 中。

10. Shortest Path algorithm

(a) Describe Bellman-Ford algorithm, which solves the single source/all destinations problem of a given graph with general weights.

< Sol >

窮舉邊 <i, j>，若 dist[i] + cost[i][j] < dist[j]，則更新最短路徑值 dist[j] = dist[i] + cost[i][j]。
重複上面步驟 n – 1 次即可獲得正確之值。

(b) Give the time complexity of Bellman-Ford algorithm according to the graph representation. Please also provide a solution to complexity reduction of Bellman-Ford algorithm.

< Sol >

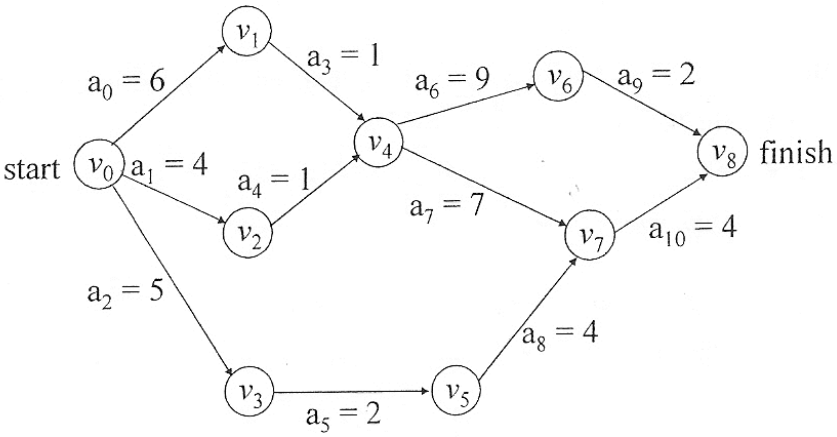
緊鄰矩陣的時間複雜度為 $O(n^2)$，而緊鄰串列的時間複雜度為 $O(e)$。
為達到減少時間複雜度的效果，可透過以下方法：

- 當目前回合沒有更新任何最短路徑值或已經執行 n – 1 回合時，便離開。
- 唯有頂點的最短路徑值更新時，周圍頂點的最短路徑值才有可能變動，因此我們可以透過 Queue 來操作，以減少冗餘的動作。

(c) Describe the well-known Dijkstra's algorithm and determine its time complexity.

- 初始化：$S = \{v_0\}\ and\ \forall i \in V, distance[i] = cost[v_0][i]$。
- 第一步：尋找一頂點 u 使得 $u \notin S$ 且 distance[u] 在 $distance[i], \forall i \notin S$ 中是最小的，並將點 u 加入 S 中。
- 第二步：$\forall v \notin S$，若 $distance[v] > distance[u] + cost[u][v]$，則 $distance[v] = distance[u] + cost[u][v]$。
- 若所有頂點的最短路徑都已經計算完畢，則結束。反之，重複以上步驟。

11. Consider the following AOE network.



(a) Obtain e(i) and l(i) of activity i, for all i.

< Sol >

Activity	0	1	2	3	4	5	6	7	8	9	10
e	0	0	0	6	4	5	7	7	7	16	14
l	0	2	3	6	6	8	7	7	10	16	14

(b) List all critical activities.

< Sol >

$a_0, a_3, a_6, a_7. a_9, a_{10}$

12. 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 t_{IO} be the I/O time, including maximum seek time, maximum latency time, and transmission time, for a block of records. In addition, nt_m represents the time to merge n records from input buffers to output buffers while t_{IS} is the time to internally sort 1000 records. What is the total time for the external sorting?

< Sol >

欲排序資料共有 48 個 Blocks，需分 12 次進行內部排序，每次處理 4 個 Blocks。

a. Read blocks, internal sort and write blocks

$$48t_{IO} \times 2 + 12t_{IS} = 96t_{IO} + 12t_{IS}$$

b. Merge runs 1 to 12 in pairs, 8 blocks

$$(8t_{IO} \times 2 + 2000t_m) \times 6 = 96t_{IO} + 12000t_m$$

c. Merge two runs of 2000 records each, 16 blocks

$$(16t_{IO} \times 2 + 4000t_m) \times 3 = 96t_{IO} + 12000t_m$$

d. Merge two runs of 4000 records each, 32 blocks

$$(32\mathrm{t_{IO}} \times 2 + 8000t_m) \times 1 = 64t_{IO} + 8000t_m$$

e. Merge two runs of 4000 records and 8000 records

$$48t_{IO} \times 2 + 12000t_m = 96t_{IO} + 12000t_m$$

f. Total

$$448t_{IO} + 12t_{IS} + 44000t_m$$

13. Answer “True” or “False” for the following statements.

(a) 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. **Ans: False**

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

(c) A stack is required for the breadth first search operation. **Ans: False**

(d) Let d, be the degree of vertex I in a graph G with |V| = n and |E| = e, then $e = \sum_{i=0}^{n-1} d_i$. **Ans: False**

(e) No edge can be in two or more bi-connected components of a graph. **Ans: True**

(f) Given a graph and a vertex, DFS and BFS may contain distinct results excluding the sequence. **Ans: True**

(g) Compared a binary search tree with a heap, the former is more suited for deleting arbitrary elements. **Ans: True**

(h) The time complexity of a deletion operation from a n-element max heap is O(n). **Ans: False**

(i) In case the least u value (in the overflow handling procedure of dynamic hashing with directories technique) is greater than the directory depth, it is required to increase both the directory size but not the number of buckets. (超出範圍)

14. Hashing

(a) What are the two principles for choosing a hashing function?

< Sol >

計算容易、碰撞情況盡量地少。

(b) If no overflow occurs, what are the two jobs which determine the time required to insert, delete, or search using hashing?

< Sol >

Hash 計算以及搜尋主桶。

15. Others

(a) Explain why a spanning tree contains exactly n – 1 edges.

< Sol >

生成樹 G'是 G 的最小子圖，使得 V(G') = V(G)且 G'為連通圖。

已知

- 生成樹也是一種樹
- n 個頂點的連通圖至少有 n - 1 個邊
- 所有只有 n - 1 個邊的連通圖都是樹

由上述可推得，生成樹有 n - 1 個邊。

(b) Show that the worst case time complexity of a quick sort is $O(n^2)$.

< Sol >

令 T(n) 為排序 n 個 Records 的時間。

最糟的情況是，每次排序時所確定的數字都剛好是最左邊的位置，如此一來整體時間複雜度為

$$\begin{aligned} T(n) &= cn + T(n-1) \\ &= cn + \big(cn + T(n-2)\big) \\ &= 2cn + T(n-3) \leq \\ &\quad \cdots \\ &= cn^2 + T(1) = O(n^2) \end{aligned}$$

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

< Sol >

以序列 (1, 5, 5, 3, 7) 為例。

(d) In a min-max heap, if the root's key is removed, the node with the smallest key value among residual nodes is either a child or grandchild of the root. Why? (微微微微超出一點點範圍)

< Sol >

Min-Max Heap 的階層性質為：

- Level 1 < Level 3 < Level 5 < … < Level n，其中 n 為奇數。
- Level 2 < Level 4 < Level 6 < … < Level n，其中 n 為偶數。

倘若原 Heap 僅有兩層，則接下來的最小鍵值在兒子裡。若有兩層以上，則接下來的最小鍵值在孫子裡。

(e) Explain how to apply dynamic programming method to solve the all pairs shortest paths problem.

< Sol >

假設 $A^k[i][j]$ 代表僅利用 $Index \leq k$ 的點計算出 i 至 j 的最短路徑。

則 $A^{-1}[i][j] = cosy[i][j]$，整體的最短路徑為 $A^{n-1}[i][j]$。

首先從矩陣 A^{-1} 開始，透過以下公式依序產生 A^0 、A^1 、A^2 、 … 、A^{n-1} 即可。

- 若不透過中間節點 k，則 $A^k[i][j] = A^{k-1}[i][j]$。
- 若透過中間節點 k，則$A^k[i][j] = min\{A^k[i][j], A^{k-1}[i][k] + A^{k-1}[k][i]\}$。

(f) Draw the tree configurations of the LL rotation of AVL trees. (超出範圍)