Data Structure - Exam I

2020/10/15

- Note: **The only acceptable programming language in your answer is C.**

1. (15%) Give the order of complexity O(•) of the following expressions.

   (a) $2^n + n^2$ $O(2^n)$

   (b) $n^{1/2} + n \log n$ $O(n^{\frac{1}{2}})$

   (c) $\sum_{i=0}^{n} x^i$ $O(x^n)$

2. (10%) Give a declaration (i.e., type definition) to the structure of the following data type Employee. (In order words, after this structure "Employee" is defined, we can set for example "Employee.Salary = 30000" and "Employee.Dependent.Spouse.Sage = 50", etc.)

Employee

Name (8 chars)	SSN (integer)	Salary (integer)	Spouse	
			Sname (8 chars)	Sage (integer)

```
① struct Employee {
     char Name[8];
     int SSN, Salary;
     Spouse Dependent;
  }
  struct Spouse {
     char Sname[8];
     int Sage;
  }
```

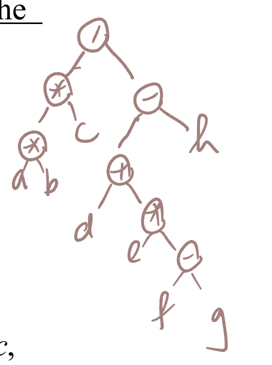
```
② class Employee {
  friend class Spouse;
  public:
     char Name[8];
     int SSN, Salary;
     Spouse Dependent;
  }
  class Spouse {
     char Sname[8]; int Sage; }
```

3. (25%) Answer the following subquestions.

   (a) Transfer the infix a*b*c/((d+e*(f-g))-h) to postfix expression and give the detailed steps of how your answers are obtained. (10%)

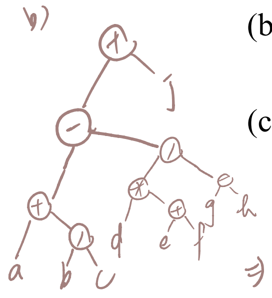
   LR Root

   ⇒ ab*c* defg-*+h-/



   (b) Transfer the postfix abc/+def+*gh-/-j+ to infix expression and give the detailed steps of how your answers are obtained. (10%)

   LR Root



   ⇒ infix = (a+b/c) - d*(e+f) / (g-h) + j ✱

   (c) What is the advantage of using a stack to evaluate a postfix expression? Explain the reason. (5%)

4. (10%) Given a string S = *a b c a b a b c a b* and a pattern P = *a b c a b c*, use the KMP algorithm to search whether P can be found in S. Detailed steps have to be given to get any score.

Step 1 先求得 P 的 prefix array = [0,0,0,1,2,3] or [-1,0,0,0,1,2]

Step 2 將 prefix array 對到上方

① abcababcab
   abcabc
   prefix = 2 ⇒ 移到 P[2]

② abcababcab
   abcabc
   prefix = 0 ⇒ 移到 P[0]

③ abcababcab
   abcabc
   prefix = 1

⇒ Not Found
但 P₈,₉ 已超過 可比對

5. (10%) The following is a C code segment. Fill in appropriate instructions in the blank spaces so that the linked list is inverted.

```
struct Node {
    int data;
    struct Node* next;
};

/* Function to invert a linked list */
static void reverse(struct Node** head_ref)
{
    struct Node* prev = NULL;
    struct Node* current = *head_ref;
    struct Node* next = NULL;
    while (current != NULL) {
        // Reverse
        [          (1)          ];   current→next = prev; ←倒置
        [          (2)          ];   next = current→next; ← 下移
        [          (3)          ];   prev= current;
        [          (4)          ];   current=next;
     }
   [          (5)          ];   head_ref = prev;
}
```

6. (30%) Answer the following subquestions about a circular queue.

(a) (10%) Define the data type of a circular queue. (Not just draw a diagram. You need to define the data type in order to get any score.)

(b) (20%) Fill in the blank spaces in the following program segments so that adding/deleting an item to/from a queue can be correctly functioned.

```
//MAX-QUEUE-SIZE is the size of the queue
element queue[MAX-QUEUE-SIZE];

/* Adding an item to a queue */
void addq(int front, int *rear, element item)
{
```

class element {
  int front, rear;
}

front & rear 是什麼?

```
  (1);
  if ((2)){
     queue_full(rear);
     return;
  }
  (3);
}

/* deleting an item from a queue */
element deleteq(int *front, int rear)
{
  element item;
  if (*front == rear)
     return queue_empty();
  (4);
  return (5);
}
```

Data Structures

Final Exam, Fall 2005

01. **(18%)** Explain the following terms:

(a)	Simple path	**(d)**	AOE networks
(b)	Articulation points	**(e)**	Spanning tree
(c)	Static hashing	**(f)**	Connected components

02. **(10%)** Consider the 2-way merge on disk. Assume there are 7200 records in disk to be sorted, using a computer with an internal memory capable of sorting at most 1200 records. Also assume that the disk I/O is with block length of 200 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 1200 records. What is the total time for the external sorting?

03. **(a)** **(5%)** Describe Prim's algorithm, which constructs minimal spanning tree of a given graph. (p.298)

(b) **(5%)** Describe the well-known Dijkstra's algorithm and determine its time complexity.

04. **(5%)** According to the definitions of depth first number and low value, describe the sufficient and necessary condition of for a vertex *u* to be an articulation point.

05. **(a)** **(2%)** With adjacency matrices representation, how to determine the degree of a vertex *i* in an undirected graph?

(b) **(5%)** 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 complexity.

06. **(10%)** Answer "True" or "False" for the following statements.

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

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

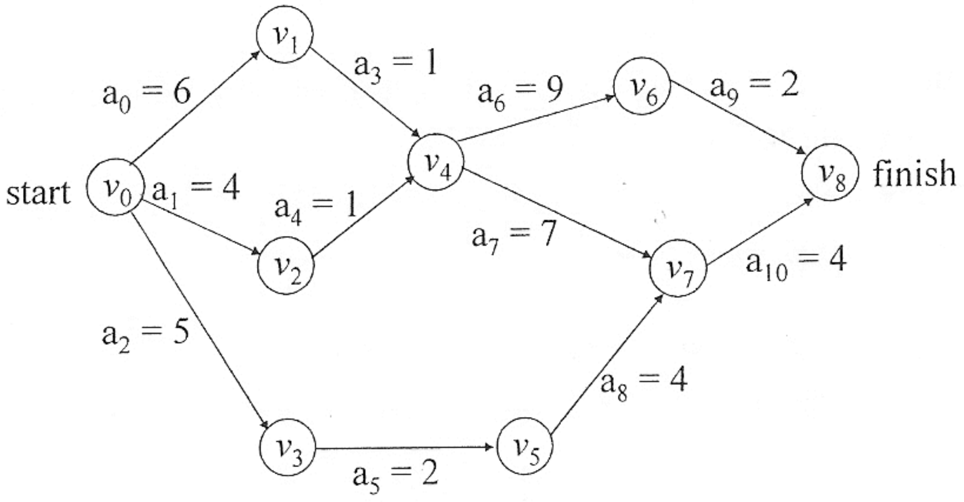
範圍內?

(c) ~~A stack is required for the breadth first search operation~~.

(d) Let d_i be the degree of vertex *i* in a graph *G* with $|V| = n$ and $|E| = e$, then

$$e = \sum_{i=0}^{n-1} d_i \, .$$

07. Consider the following AOE network. (p.310)



(a) **(10%)** Obtain *early(i)* and *late(i)* of activity *i*, for all *i*.

(b) **(3%)** List all critical activities.

08. **(a)** **(3%)** Explain why a spanning tree contains exactly $n-1$ edges. (p.278)

(b) **(5%)** Show that the worst case time complexity of a quick sort is $O(n^2)$.

(c) **(3%)** Heap sort is not stable. Give an example of an input list in which the order of records with equal keys is not preserved.

(d) **(3%)** 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?

09. **(10%)** We can use the tree structure for set representation. In this application, *union* and *find* are the minimal operations; the former 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?)