編號： 189

國立成功大學 106 學年度碩士班招生考試試題

系　　所：電腦與通信工程研究所

考試科目：資料結構

考試日期：0214，節次：2

※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

一、 複選題: (20 分　，每題 5 分，全對才給分)

1. Which of the following statement(s) is (are) true?

(A) The time complexity of a breath-first algorithm is O(V+E) which is the fastest search algorithm.

(B) The time complexity of Dijkstra algorithm is O(E lg V) if it is implemented by an array .

(C) There may exist back edges in a spanning tree generated by a breath-first algorithm.

(D) The time complexity to delete the maximum value in a heap is O(1).

2. Which of the following statement(s) is (are) true (assume each set contains an unique element initially and it is implemented by the disjoint-set forest while the weight rule is applied)?

(A) In the worst case, find an element in a set of size n take Θ(logn).

(B) The weight rule is to make the taller tree as a subtree of the shorter tree during union.

(C) p unions and q finds can be done in O(p+q).

(D)The kruskal algorithm is implemented by set and the data structure is used to check if the two nodes of a new found edge will form a cycle.

3. Which of the following statement(s) is (are) true?

(A) Merge sort is faster than the insertion sort algorithm in all conditions.

(B) The expected time complexity of the quick sort algorithm is O(nlogn) when using RANDOMIZED-PARTION.

(C) The complexity of a comparison based algorithm cannot be faster than O(nlogn).

(D) A LSD Radix sort is not a stable sort.

4. Which of the following statement(s) is (are) true?

(A) If n1 and n2 are the left child and right child of a node n in a max heap, the value of n2 cannot be larger than n2.

(B) It takes O(logn) to search a maximum value in a binary search tree if the tree has n nodes.

(C) A binary search tree is usually implemented by a complete binary tree.

(D) The time complexity to perform Max Heapify can be as fast as O(n).

國立成功大學 106 學年度碩士班招生考試試題

系　　所：電腦與通信工程研究所

考試科目：資料結構

考試日期：0214，節次：2

二、簡答題: (60 分)

1. (20 **pts**) Give a table which characterizes the priority of the operators.

priority	ISP	ICP
(		0
!	1	1
* / %	2	2
+ -	3	3
=	4	4
(	5	

The expression is (A + B) *(C – !D / E) % F

a. (5 **pts**) Show the equivalent postfix expression if we use a stack to evaluate the postfix expression in C++ from left to right.

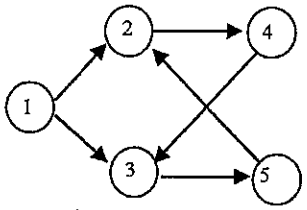
b. (5 **pts**) Based on a, please show the status of the stack in each time before you pop data, how many pop operations you have to perform.

c. (7 **pts**) Draw a binary tree representation which corresponds to the sequence of its infix expression.

d. (3 pts) Please show the results of the expression 5*2 +30 -20 -80/5 if we reverse the priority of the operators * / with the priority of the operators + - and the evaluation of operators of the same priority will proceed from right to left.

2. (15 **pts**) **Transitive closure graph.**

According to the graph G shown in the following:



a. (1 **pts**) Use an adjacency matrix to represent the graph.

b. (6 **pts**) Find the transitive closure matrix A^+ and the reflexive transitive closure matrix A^* of G.

c. (2 **pts**) Explain the meaning of A^+.

d. (2 **pts**) Explain the meaning of A^*.

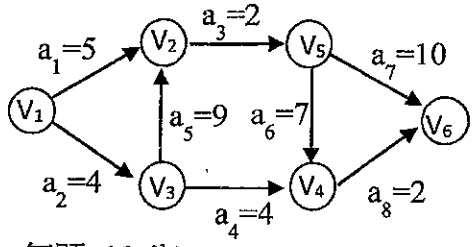
e. (4 **pts**) If we use a matrix A^0 to represent the distances between any two vertices, where t $A^0(i, j)$ denotes the shortest distance between two vertices i and j, Floyd-Warshall algorithm can find the shortest path between any two vertices by compute a new matrix A^k from A^{k-1}. Please list the general recursive equation and explain the equation shortly (hint: what does k represent in the equation?).

3. (25 **pts**) **Active On Edge (AOV) network.**
   a. (2 **pts**) Please show the adjacency list of the figure (assume the sequence of nodes in a list are arranged according to their numbers in increasing order).
   b. (2 **pts**) Find a topological sequence of nodes according to the data structure obtained in a.
   c. (6 **pts**) Find the earliest even occurrence time $ee(i)$ and the latest even occurrence time $le(i)$ for each even i and show the results in a table according to the sequence of the index of even i.
   d. (8 **pts**) Find the earliest time, $e(i)$, and latest time, l(i), for each activity and shows the result in a table according to the sequence of the index of activity i.
   e. (1 **pts**) What is the earliest time when the project can be finished?
   f. (2 **pts**) Please show the slack of each activity.
   g. (2 **pts**) Which activities are critical and why they are critical?
   h. (2 **pts**) Please show the activities which can result in reduction of the project length if they speed up.

$e(i) = ee[k] = \max_{<j,k>\in E} \{ee[j] + \text{duration\_on\_}<j, k>\}$

$l(i) = le[l] - \text{duration\_of\_}a_i$

$= \min_{<l,j>\in E} \{le[j] - \text{duration\_on\_}<l, j>\} - \text{duration\_of\_}a_i$



三、填充題 (20 分，每題 10 分)

1. Give the following algorithm whose input is a graph G, an edge weight w, and a vertex r in G.

```
WeightGraph(G, w, r)
 for each u ∈ V(G)
     key[u] = ∞
 Insert all nodes into Q;      // Q denotes a priority queue.
 key[r] = 0;
 while (Q is not empty)
     u = ExtractMin(Q)
     for each neighbor v of u and v ∈ Q
         if (_____A_____)
             _____B_____
```

a. (4 **pts**) Please list the answers of A and B if the algorithm is to find the minimum spanning tree.
b. (4 **pts**) Please list the answers of A and B if the algorithm is to find the single source shortest tree.
c. (1 **pts**) Please give the time complexity in term of V and E.
d. (1 **pts**) Please give the time complexity if we replace Q by an array A.

國立成功大學 106 學年度碩士班招生考試試題

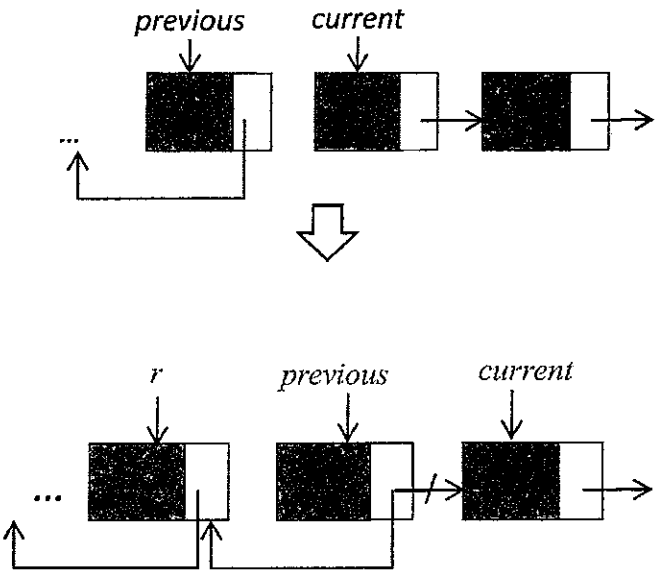
系　　所：電腦與通信工程研究所

考試科目：資料結構

考試日期：0214，節次：2

2. (10 **pts**) Please reference the figure in the bellow to complete the pseudo code of the reverse which reverses the sequence of nodes in a list.

```
class Chain; // forward declaration
class ChainNode {
friend class Chain;
public:
    ChainNode (int element = 0, ChainNode* next = 0)
        // 0 is the default value for element and next
        {data = element; link = next;}
private:
    int data;
    ChainNode *link;
};
class Chain { // true definition
    ChainNode *first; // has a ChainNode pointer
public:
    void Reverse(); // reverse the sequence of nodes
                    // in a list
};
```




```
template <class T>
void Chain<T>::Reverse()
{ // A chain is reversed so that (a1, ..., an) becomes (an, ..., a1)
    ChainNode <T> *current = first,
                  *previous = 0; // previous trails current
    while (___A___) {
        ChainNode <T> *___B___;
        ___C___;
        ___D___;
        ___E___;
    }
    first = previous;
}
```