淡江大學97學年度碩士班招生考試試題

129-1

系別：資訊管理學系　　科目：資　料　結　構

本試題共 3 頁，7 大題 1/3

1. Define the following terms. (16%)
(a)quadratic probing (b) AVL trees (c)priority queues (d) the software life cycle

2. Bubble Sort (12%)
(a) Complete the following C++ codes of bubble sort. The user can sort an integer array a[] with n elements in ascending order by calling `bubbleSort(a,n)`.
(b) Analyze the time complexity of bubble sort.

```
void swap(int &a, int &b) ;
void bubbleSort(int a[], int n) {
   bool sorted ;
   for (int pass=1; pass<__(1)__; pass++) {
        sorted =__(2)__ ;
        for (int i = 0 ; i<__(3)__; i++) {
             if (__(4)__) {
                  __(5)__;
                  sorted =__(6)__;
             }
        }
        if (sorted) __(7)__;
   }
}
```

3. Mergesort (13%)
(a) Complete the following C++ codes of mergesort. The user can sort an integer array a[] with n elements in ascending order by calling `mergesort(a, 0, n-1)`. Note that the codes is a recursive implementation of mergesort.
(b) Analyze the time complexity of mergesort.

```
void mergesort(int a[], int first, int last) {
   if (first < last) {
        int mid = (first+last)/2 ;
        _____(1)_____;
        _____(2)_____;
        merge(a, first, mid, last) ;
   }
}
```

```
void merge(int a[], int first, int mid, int last) {
     int temp[MAX_SIZE] ;
     int first1 = first, last1=mid ;
     int first2 =mid+1; last2 = last ;
     int index =first ;
     while (__(3)__) {
          if (a[first1]<a[first2])
               __(4)__;
          else
               __(5)__;
          index++ ;
     }
     // other codes are omitted intentionally
}
```

本試題雙面印製

◀注意背面尚有試題▶

淡江大學97學年度碩士班招生考試試題

系別：資訊管理學系　　科目：資　料　結　構

本試題共 3 頁，7 大題 2/3

4. Linked List (15%)

(a) Complete the following C++ codes of linked list operations by filling the blanks. The list_print() function uses the recursive concept to print out the content of a linked list. The user can print a list L by calling `list_print(L.head)`. The list_intersect() function calculates the intersection of two linked lists, e.g., if the contents of list L1 and L2 are (2, 3, 5, 7, 9) and (11, 3, 7, 10) respectively, calling `list_intersect(L1,L2)` would return a new list L3 containing (3, 7). (10%)

(b) Compare the advantages and disadvantages of linked lists with those of array lists when considering them as the implementation of the ADT List. (5%)

```
struct node {
    int info ;   node *next ;
} ;
struct list { // list without a dummy header node
    node *head ;
    list() { head = NULL; }
} ;
void list_append(list& L, int info) {
    // append a node to the tail of L
    // and let the content of the node be info.
    // The codes are omitted for simplicity.
}
```

```
void list_print(node *ptr) { // recursive form
    ____(1)____;
    cout << ptr->info <<endl ;
    ____(2)____;
}
list list_intersect(list& L1, list& L2) {
    list L3 ;
    node *ptr = L1.head ;
    while (____(3)____) {
        node *ptr1=__(4)__ ;
        while (__(5)__) {
            if (ptr->info == ptr1->info) {
                ____(6)____;
                break ;
            }
            ___(7)___;
        }
        __(8)__;
    }
    return L3 ;
}
```

5. BST and tries (16%)

(a) construct a binary search tree by inserting the following key values sequentially.

"Quad", "Tom", "Peter", "Tommy", "Sam", "Okay", "Otter", "Ottur"

(b) present the results of preorder and postorder traversals in the tree constructed in (a). (assume the task of traversal is to print the key value of each node)

(c) build a trie for the key values shown in (a).

(d) compare the efficiency of binary search trees with that of tries.

6. Hashing (12%)

(a) Insert the following keys into a hash table (table size=13) and adopt double hashing as the collision resolution technique.

11, 105, 41, 27, 81, 110

The hash value of each key is calculated as follows:

$h(key) = h_1(key) + i*h_2(key)$, i: the times of collision

$h_1(key) = key \bmod 13$, $h_2(key) = 7-(key \bmod 7)$

In addition to the calculation process, you should present your final answer by sketching the contents of the hash table in pictorial forms.

(b) How can the double hashing technique probe all the table locations?

7. Refer to the following adjacency list representation of a graph G1, (16%)

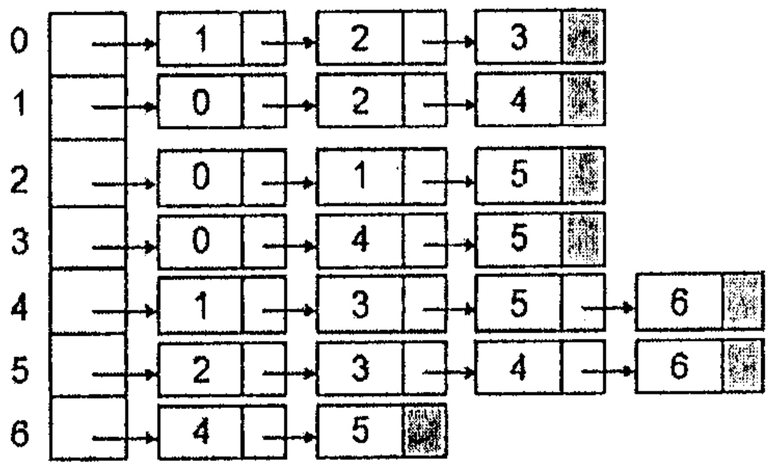
(a) Sketch G1 in a pictorial form. (3%)

(b) Show the visiting order of depth first search in G1 if node 0 is the start node.

(c) Similar to (a), but using node 4 as the start node. (3%)

(d) Refer to graph G2 and use Kruskal algorithm to find the minimum cost spanning tree. You should present the process of how to add edges to the spanning tree, but not just show the resultant spanning tree. (7%)

G1



G2