國立彰化師範大學115學年度碩士班招生考試試題

系所：數學系(選考丁)、統計資訊研究所(選考丁)　　科目：資料結構

☆☆請在答案紙上作答☆☆　　

一、(50%)請寫出下列程式碼的執行結果(共 5 題，每題 10 分)：

(1)

```
#include <stdio.h>
#include <limits.h>
int main() {
    int prices[] = {7,1,5,3,6}, n = 5;
    int minPrice = INT_MAX, maxProfit = 0;
    for (int i = 0; i < n; i++) {
        if (prices[i] < minPrice) {
            minPrice = prices[i];
        } else if (prices[i] - minPrice > maxProfit) {
            maxProfit = prices[i] - minPrice;
        }
    }
    printf("%d", maxProfit);
    return 0;
}
```

(2)

```
#include <stdio.h>
int main() {
    int t[] = {7,1,5,3,6}, n = 5;
    int r[5] = {0};
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (t[j] > t[i]) {
                r[i] = j - i;
                break;
            }
        }
    }
    for (int i = 0; i < n; i++){
        printf("%2d", r[i]);
    }
    return 0;
}
```

(3)

```
#include <stdio.h>
void gcd(int a, int b) {
    printf("%d %d\n", a, b);
    if (b == 0)
        return a;
    return gcd(b, a % b);
}
int main() {
    int n1 = 74, n2 = 13;
    gcd(n1, n2);
    return 0;
}
```

(4)

```
#include <stdio.h>
int main() {
    int a[] = {7,1,5,3,6}, n = 5;
    int flag;
    do {
        for (int j = 0; j < n; j++){
            printf("%d",a[j]);
        }
        printf("\n");
        flag = 0;
        for (int i = 0; i < n - 1 ; i++){
            if (a[i] > a[i+1]){
                int t = a[i];
                a[i] = a[i+1];
                a[i+1] = t;
                flag = 1;
            }
        }
    } while (flag == 1);
    return 0;
}
```

(5)

```
#include <stdio.h>
int main() {
    int arr1[] = {1,2,5,6}, n1 = 4;
    inr arr2[] = {3,4,7}, n2 = 3;
    int i = 0, j = 0;
    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j])
            printf("%d ", arr1[i++]);
        else
            printf("%d ", arr2[j++]);
    }
    return 0;
}
```

二、Consider an initially empty Binary Search Tree (BST). Insert the following distinct keys into the BST in the given order using the standard BST insertion rule:

- Keys smaller than a node go to the left subtree, keys larger go to the right subtree.
- No balancing is performed during insertion.

Insertion order: 30, 15, 50, 10, 20, 40, 60, 5, 13, 18, 23

Please answer the following:

a. Draw the final BST after all insertions are completed. (4%)

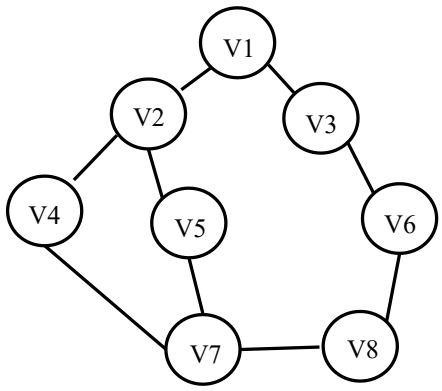
b. When searching for key = 23, list the visited node sequence (in order) and the total number of key comparisons. (Assume one comparison per visited node: compare target key with current node's key.) (3%)

c. State the best-case and worst-case time complexity of BST search in terms of n. Briefly explain how AVL tree improves the worst-case. (3%)

三、The following is an undirected graph G with vertices V1 to V8. Neighbor order rule: when there are multiple adjacent unvisited vertices, always choose them in increasing vertex index order.

a. Starting from V1, perform Depth First Search (DFS) and write the vertex visiting order. (5%)

b. Starting from V5, perform Breadth First Search (BFS) and write the vertex visiting order. (5%)



四、Consider an initially empty B-tree with minimum degree t = 2.

- Each node can have at most 2t – 1 = 3 keys.
- Except for the root, each node must have at least t – 1 = 1 key.
- When a node is split, the median key is promoted to the parent (standard B-tree insertion rule).

Insert the keys in order: 10, 20, 5, 6, 12, 30, 7, 17

Please answer the following:

a. Draw the final B-tree after all insertions. (Keys inside each node must be sorted.) (5%)

b. Indicate which insertions cause a split, and for each split, specify the key that is promoted to the parent. (5%)

五、The hash function is: h(k) = k mod 11, Insert the keys in order: 22, 1, 12, 33, 24

Please answer the following:

a. Compute the hash index $h(k)$ for each key. (5%)

b. Use separate chaining to handle collisions. Draw the final hash table (show the chain in each bucket). (Assume new keys are appended to the tail of the chain to keep insertion order.) (5%)

六、Given the postfix expression: 7 2 3 ＊ − 4 5 + ＊

Assume all operations are integer arithmetic. Please answer the following:

a. Compute the final value of this postfix expression. (5%)

b. Show the key stack push or pop process (4-6 steps are sufficient; focus on the operator steps). (5%)