國立嘉義大學 99 學年度

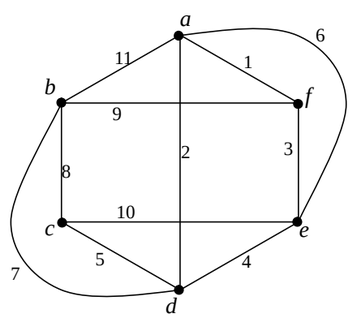
資訊工程學系碩士班（甲組）招生考試試題

科目：數學

一、If you have stamps of two different denominations, 5 dollars and 7 dollars, you can make up exactly any postage of n dollars or more using stamps of these two denominations.
(1) What is the minimum of n? (5%)
(2) Please prove it. (10%)

二、Please give a grammar that specifies each of the following languages:
(1) $L=\{a^{3i}(bc)^{j} \mid i \geq 1, j \geq 1\}$ (5%)
(2) $L=\{x \mid x \in \{a, b\}^{\star}$ and x does not contain two consecutive a's$\}$(Notice that the notation $S^{\star}$ denotes the set of all sequences of letters from Set S.) (5%)

三、Determine a minimum spanning tree for the following graph. (10%)

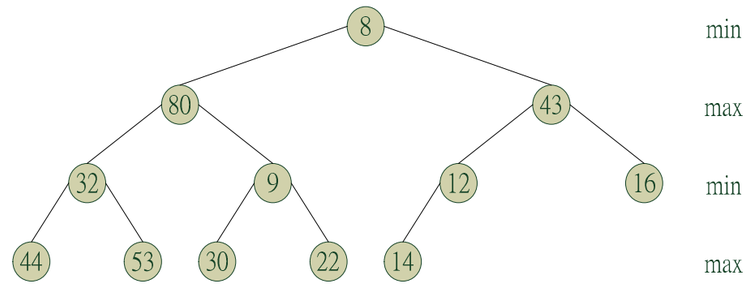


四、Solve the following recurrence relations:
(1) $a_{n+1}-2a_n=5,\ n\geq 0,\ a_0=1.$ (8%)
(2) $a_{n+2}-5a_{n+1}+6a_n=4n,\ n\geq 0,\ a_0=5,\ a_1=10.$ (7%)

五、Write the postfix form of the following expressions: (10%)

(1) (A + B) × D + E / (F + A × D) + C
(2) not (A and not ((B < C) or (C > D))) or (C < E)

六、The following figure is a min-max heap h_1 with 12 elements. The value in each node is the key of the element in that node. Show the min-max heap h_2 after we insert the element with key 6 into h_1. (10%) Next, suppose we wish to insert the element with key 92 into h_2, and show the min-max heap h_3 after inserting 92. (10%)



七、Procedure ***QuickSort*** shown in the following sorts elements 1 through n of a list ***list***, and the procedure invocation is ***QuickSort(list, 1, n)***. In addition, Procedure ***InterChange(x, y)*** performs $t = x; x = y; y = t$. Suppose that the input list has 10 records with keys (26, 5, 37, 1, 61, 11, 59, 15, 48, 19). Give the full status of the input list at each call of ***QuickSort***. (20%)

```
Procedure QuickSort( List list, int s, t) {
    int i, j, pivot;
    if s < t {
        i = s;    j = t +1;    pivot = list[s].key;
        do {
            do {
                i +=1;
            } while (list[i].key < pivot);
            do {
                j -= 1;
            } while (list[j].key > pivot);
            if (i < j)
                InterChange(list[i], list[j]);
        } while (i < j);
        InterChange(list[s], list[j]);
        QuickSort(list, s, j -1);
        QuickSort(list, j + 1, t);
    }
}
```