M.TECH/CSE/1ST SEM /CSEN 5104/2015
2015

Advanced Problem Solving & Programming
(CSEN 5104) 21

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following: **10 x 1=10**

(i) A table TAB stores 100 distinct integers in sorted order. It is found that the average time for the successful search of a key in TAB using the binary search algorithm is 2 secs. The number of items in TAB is increased to 1000 distinct integers. Assuming TAB is still in sorted order, approximately how much time do you expect it to take now on an average for a successful search of a key using the same binary search algorithm?

(a) 2 secs (b) 3 secs (c) 5 secs (d) 10 secs.

(ii) We enter n random integers one by one into a hash table TAB of 511 entries using open addressing with linear probing. It can be assumed that $100 \leq n \leq 400$. Then the average time taken to find a key in TAB is

(a) proportional to n
(b) proportional to n^2
(c) constant and independent of n
(d) proportional to log n.

(iii) A hash table TAB of size m containing n integer entries, where $0 < n < m$, has been created using open addressing with linear probing. We now realize that we need to find both the largest and the smallest entries in TAB. We will be able to do this in time that is

(a) proportional to n
(b) proportional to m;
(c) independent of both n and m
(d) proportional to $(n^2 + m)$.

(iv) The total number of distinct (i.e., incongruent) right-angled triangles in which the length of each side is an integer and the length of the hypotenuse is <= 15 is

(a) 3 (b) 4 (c) 5 (d) 6.

(v) The total number of distinct (i.e., incongruent) triangles in which the length of each side is an integer and the perimeter of the triangle is <= 15 is

(a) 8 (b) 7 (c) 6 (d) 5.

(vi) The six letters a, a, b, c, c, d are to be arranged in linear order (i.e., permuted), but it is specified in addition that 'a' must be the first letter in the arrangement, and 'c' must be the last letter in the arrangement. In how many different ways can we arrange the letters?

(a) 24 (b) 48 (c) 120 (d) 180.

(vii) The depth (D_n) of a complete binary tree having n nodes (assuming depth of the root is 1) is given by

(a) $D_n = \lceil \log_2 n \rceil$ (b) $\lceil D_n = \log_2(n+1) \rceil$

(c) $D_n = \lceil \log_2 n + 1 \rceil$ (d) D_n = none of these.

(viii) A binary tree T has 10 internal nodes and an appropriate number of leaf nodes. Then the total number of edges in T is

(a)40 (b) 30 (c) 20 (d) 10.

(ix) An array A of length 80 contains a sequence of open '(' and closed ')' parentheses. We have been asked to check whether the sequence of parentheses in A is well-formed (i.e., balanced) using a pushdown stack. To solve this problem we need a stack of size at least

(a)21 (b)31 (c) 41 (d) 81.

(x) A binary tree T having 15 nodes is stored in a 15x15 matrix M in the following form: M[j,k] = 1 if nodes j and k are connected by an edge in T, and M[j,k] = 0 otherwise. Then the number of 0's in matrix M is

(a)15 (b) 125 (c) 197 (d) 211.

Group – B

2. Answer the parts below giving reasons in each case.

i) How many distinct binary search trees containing the three keys 5, 7 and 8, when traversed in inorder, will all yield the traversal sequence '5 7 8' ?

ii) What is meant by the balance factor of a node in a binary tree? When is a binary tree said to be balanced? Consider all binary trees that have five nodes. How many of these are balanced?

iii) A binary search tree contains the seven keys 10, 20, 30, 40, 50, 60, 70. If T has height 3, which keys can possibly occur at the root of T? If T has height 5, which keys can possibly occur at the root of T?

(4 + 4 + 4) = 12

3. Write a program in C to solve the following problem:

An array TAB contains n positive integers, not all of which are distinct. You are required to create another array ARR of m entries, m <= n, which will contain only distinct entries from TAB. If an integer occurs k times in TAB where k > 1, only the leftmost occurrence should be included in ARR, the remaining k-1 occurrences should be eliminated. Thus at the end there will be distinct elements in ARR, and these elements will retain the same left to right order as in TAB. For example, if n = 10 and TAB contains

45 37 80 7 37 80 37 21 45 60

then your program will output m = 6 and the ARR entries

45 37 80 7 21 60.

You will be provided as input the value of n and the integers that will enter TAB. You will output m and the entries in ARR as output.

12

Group – C

4.(a) Construct a B-tree of order 5 with the following keys:

a, g, f, b, k, d, h, m, j, e, s, i, r, x, c, l, n, t, u, p

Now delete a node with key 'r' from the tree just constructed.

(b) How does hash search differ from Binary or Linear search? Suppose that a hash table contains HASHSIZE = 13 entries indexed from 0 through 12 and that the following keys are to be mapped into the table:

10, 100, 32, 45, 58, 126, 3, 29, 200, 400, 0

Determine the hash addresses and find how many collisions occur when these keys are reduced by %HASHSIZE.

(5+2)+5=12

5. A character string hexnumber is supplied as input, with 10-15 being represented by the letters A-F. A C program has to be written that will spell out each digit in words. For example, for the input the output will be as shown below:

hexnumber	A	6	5	B
digits in words	ten	six	five	eleven

The output will then be a string of the form: "ten six five eleven".

You have been asked to design (and then write) the program. Describe precisely in words in a step by step manner your design for the program that will accomplish this task. (You do not need to write the program in C, only to explain how it will work.)

12

Group – D

6.(a) Insert the following 12 keys into a hash table H1 of size 15 (indexed from 0 through 14) using the open addressing scheme with linear probing:

20, 16, 200, 10, 100, 34, 45, 57, 136, 3, 29, 88

Show the final hash table and determine the average number of probes for a successful search.

(b) The same set of 12 keys are to be inserted into another hash table H2 of size 15 using the open addressing scheme with double hashing. The two hash functions are $h_1(k) = k \bmod 15$ and $h_2(k) = k \bmod 13$. Again show the final hash table and determine the average number of probes for a successful search.

6+6=12

7.(a) The recursive functions f(.) and g(.) both take the positive integers as domain and the integers as range, and are defined as follows:

$f(n) = 1$ when $n = 1$

$f(n+1) = f(n) + 2n + 1$ when $n > 1$

$g(n) = 1$ when $n = 1$

$g(n+1) = -g(n) + (-1)^n.2$ when $n > 1$

The function h(.) also takes the positive integers as domain and the integers as range, and is defined as follows:

h(n) = f(g(n)) when n >= 1

Explain what is being computed by h(n) and compute the value of h(5).

(b) Write a program in C that, when supplied a positive integer n as input, computes and outputs h(n).Make use of recursion to find the values of f(.) and g(.) as needed.

4 + 8 = 12

Group – E

8. Answer the following questions with reference to the C++ programming language, giving appropriate illustrative examples in each case.

i) In what way does the use of class definitions in a program help to reduce the possibility of occurrence of software faults?

ii) In which situations would it be an advantage to declare a data member of a class to be static?

iii) How do we gain by the use of constructors? Is it possible to avoid the use of explicit constructors in a program?

iv) Describe a situation in which the inheritance feature plays a useful role in a program?

(3 + 3 + 3 +3) =12

9. Write a program *reversible* in C or C++ to solve the following problem:

An array *A* contains a string of *n* characters, where *n* is a positive integer, each entry in the array being '*a*', '*b*', '*c*' or '*d*'. The character string will be called *reversible* if the following condition holds. When all occurrences of '*d*' in the string are ignored, the sequence of characters reads the same forwards as backwards. For example, suppose the string is the sequence

dbaddccdddadcddcadb

Ignoring the symbol '*d*' we get *baccaccab* which reads the same forwards as backwards, so this string is reversible. But the string *cbdabddbadacdd* is not reversible. Of course, all four characters '*a*', '*b*', '*c*' and '*d*' need not be present in a given string. It can be assumed that no other characters are present in the string.

Given a string of the indicated form, your program should output a message stating whether or not the string is reversible.

12