International Institute of Information Technology, Bangalore.
CS 501 Data Structures and Algorithms.
EndTerm Exam: December 11, 2014.

1. m **unit** jobs need to be scheduled on $n(m > n)$ machines. All the jobs require same amount of processing time, but a job cannot be processed on any machine, it can only be processed on a subset of machines. You are given p_{ij}, p_{ij} is TRUE if the ith job can be processed on jth machine, FALSE otherwise. Design an efficient algorithm to schedules the jobs on these machines to minimize the maximum number of jobs processed on a machine. **(4 marks)**

2. You are given a sequence of n real numbers. Design an $O(\log n)$ algorithm, which finds a number whose rank is greater than $n/4$ with probability greater than $1 - 1/n$. **(3 marks)**

3. Govt plans to build hospitals along a high way from A to B, they have identified n locations on the highway where they can build hospitals. Let $x_1 < x_2 < \ldots x_n$ be the distances of these locations from B. They already have one hospital each at A and B. They want to build minimum number of hospitals, but ensure that the distance from any location on the highway to the nearest hospital is at-most k kms.

   Design an efficient algorithm to solve this problem. Prove it is correct and analyze your algorithm. **(4 marks)**

4. $G = (V, E)$ be a graph with positive edge weights. You are given a set of warehouse vertices say $W \subset V$. Design an $O(E \log V)$ algorithm to compute the shortest distance from every vertex to the nearest warehouse vertex.**(3 marks)**

5. We say that a graph is a near-tree if it is connected and has at most $n + 8$ edges, where n is the number of vertices in the graph. Give an $O(n)$ algorithm to compute the minimum spanning tree of a near-tree. **(3 marks)**

6. Suppose a minimum spanning tree T of a graph G has been already computed. Give an efficient algorithm to update the minimum spanning tree, if a new vertex and (say k, weighted) edges incident to the newly added vertex are added to G. What is the complexity of your algorithm ? **(3 marks)**

7. Consider the weighted version of job scheduling problem, in which each job comes with a weight w_i, start time s_i and end time e_i. Here we would like to maximize the total weight of the jobs that are scheduled. Only one job can be scheduled at a time on the machine.

   Suppose jobs $\{[0, 6], [2, 10], [9, 15], [7, 18]\}$ have weights $2, 4, 6, 7$. Then scheduling the first and the last job gives a weight of $2 + 7 = 9$, which is maximum.

   Design an efficent algorithm for this problem.**(6 marks)**

8. Given a binary matrix, design an efficient algorithm to find the maximum size sub-matrix with all 1's. What is the complexity of your algorithm ?**(6 marks)**

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Please go through the following pseudo code and work the PROBLEM given at the end. It is fun to make a dry run of this problem. This problem is an illustration for 'BACKTRACKING' algorithms. Also trains you in recursive logics.

```
int turnpike( int x[],DistSet D, int N)
    {
        x[1] = 0;
        x[N] = DeleteMax (D);
        x[N-1] = DeleteMax (D);
        if ( (x[N] – x[N-1])  is in the set D)
            { Remove ( x[N] – x[N-1], D);
              return Place(X, D, N, 2, N-2);
            }
        else
            return false;
    }
int Place(int X[], DistSet D, int N, int left, int right )
{   int Dmax, Found = False;
    if ( D is Empty )
        return true;
    Dmax = FindMax( D );
    if ( AbsVal( X[j] – Dmax) is in D for all 1 <= j < left and right < j
<= N )
        { X[right] = Dmax;
          for ( 1 <= j < left, right < j <= N )
                Remove ( AbsVal( X[j] – Dmax), D );
          Found = Place( X, D, N, left, right – 1);
          if ( !Found )      /* note: !Found means 'not Found' */
           { for( 1 <= j < left, right < j <= N )
                Insert( AbsVal( X[ j ] – Dmax), D );
           }
        }
        if ( !Found &&  ( AbsVal( X[N] – Dmax – X[ j]) is in D
                          for all 1 <= j < left and right < j <= N ))

            { X[ left] = X[N] – Dmax;
```

```
        for ( 1 <= j < left, right < j <= N )
          Remove ( AbsVal( X[ N] – Dmax – X[ j] ), D );
        Found = Place( X, D, N, left + 1, right );
     if ( !Found )
      { for( 1 <= j < left, right < j <= N )
          Insert( AbsVal( X[ N ] – Dmax – X[ j]), D );
      }
     }
     return Found;
}
```

NOTE : The following is description of various functions called in the above pseudo code.

1. DistSet D : it is set of integers- repetition allowed. Eg : D = {2, 2,3,3,3,5}
2. AbsVal (Z) – returns absolute value of integer Z.
3. FindMax(D) – returns the maximum in the set D.
4. DeleteMax(D) – deletes the max element from D and *returns* the max as output.
5. Remove(x,D) – removes the element x from the set D.
6. Delete(x, D) – Deletes one entry of x from D. Returns the new set D.
7. Insert (x, D) – inserts the value x into the set D.
   Example for above functions. Let D = {2,2,3,3,3,5}
   then FindMax(D) = 5
   Delete(3,D) gives D = {2,2,3,3,5}
   Insert(6,D) gives D = {2,2,3,3,3,5,6}
   DeleteMax(D) gives D={2,2,3,3,3}
   Remove(5,D) gives D={2,2,3,3,3}

PROBLEM

Take the input as below.

N = 6

The set D = { 1,2,2,2,3,3,3,4,5,5,5,6,7,8,10 }

for the function 'turnpike' and make a dry run of the above pseudo code

WRITE DOWN THE output array (X[I], I = 1,6).

International Institute of Information Technology, Bangalore.
CS501 Data Structures and Algorithms.
Mid Term Exam, 30 September 2013.

1. Which of the following statement is true.

   Justify your answer. **(5 marks)**

   (a) There exists a constant $n_0 \geq 1$, such that, for every $n \geq n_0$, there is an array of n elements on which insertion sort runs faster than merge sort.

   (b) In-order traversal and Post-order traversal uniquely determines a binary tree.

   (c) Pre-order traversal and Post-order traversal uniquely determines a binary tree. F

   (d) It is possible to convert a binary heap to a binary search tree in linear time. F

   (e) Given a sequence of integers in increasing order, a balanced binary search tree can be built in linear time. T

2. You are given a $n * m$ matrix of numbers in which entries of each row is sorted in increasing order from left t o right and entries of each column is sorted in increasing order from top to bottom. Design an $O(n + m)$ time algorithm to search for an element x in the matrix. **(3 marks)**

3. An element in a given sequence $a_1, a_2, \ldots a_n$ is said to be a majority element, if it repeats at least $n/4 + 1$ times. Design a linear time algorithm to decided if a given sequence has a majority element.

   (2 marks)

4. What is the minimum number of elements that need to be hashed into a hash table of size m with chaining in oder to make sure that at least one hash location has a chain of length n ? **(2 marks)**

5. We would like to store a sparse matrix (a matrix populated primarily with zeros) of integers with n rows and n columns in one of the two methods, where the first method is to store only the non-zero integers in "list of lists" and the second method is a array of size n by n. Let there be r non-zero integers in the matrix. Determine the amount of

space used by each method to store the sparse matrix. Which method takes less space when $r = n^2/2$, assuming that space needed to store both an integer and an integer pointer is 1 byte. **(2 marks)**

6. The preorder traversal of an AVL tree is $44, 17, 32, 78, 50, 48, 62, 88$. **(3 marks)**

   (a) Give the postorder traversal of the AVL tree after inserting 54.

   (b) Give the postorder traversal of the AVL tree after deleting 32.

7. Write a program to enumerate all the keys k, such that $a.data \leq k \leq b.data$ in a binary search tree. The enumerate function should be implemented in $O(m)$, where m is the number of keys in the output. **(4 marks)**

   struct node {int data; struct node *left, *right; }

   void enumerate (struct node *a, struct node *b)

8. We would like to maintain a dynamic set S of numbers that supports the operation $MINGAP$, which gives the magnitude of the difference of the two closest numbers in S. For example, if $S = \{4, 18, 56, 15, 24, 36\}$, then $MINGAP(S)$ returns 3, since 18 and 15 are the closest numbers in S. Similarly, $MAXGAP$, gives the maximum of the difference of the two numbers in S. For example, if $S = \{4, 18, 56, 15, 24, 36\}$, then $MAXGAP(S)$ returns 52.

   Design a data structure that supports $INSERT, DELETE, SEARCH$ operations in $O(\log n)$ time and $MINGAP, MAXGAP$ operations in $O(1)$ time. **(5 marks)**

9. Design a data structure on a set S of integers which supports the following two operations so that any sequence of n operations runs in $O(n)$ time. **(3 marks)**

   - Insert (S, x): Insert x into set S.
   - Delete-Larger-Half (S): Deletes the largest $\lceil S/2 \rceil$ elements from S.

10. Height of a Fibonacci-heap is defined as maximum height of any node in the Fibonacci-heap. Given a positive integer n, give a sequence of Fibonacci-heap operations that creates a Fibonacci-heap of height n. **(3 marks)**