International Institute of Information Technology, Bangalore
CS101 Algorithms Mid Term Exam , 4 October 2011

Answer any FIVE of the following questions.

1. Given a sorted array of distinct integers $A[0] < A[1] < \ldots A[n-1]$, design an $O(\log n)$ algorithm for the following

   (a) Decide whether there is an index i such the $A[i] = i$.

   (b) Given x and y, find the number of integers in the given array which are strictly greater than x, but strictly smaller than y.

2. Let A, B and C be three sequence of n integers each. Design an $O(n^2)$ algorithm to determine if there are three integers $a \in A, b \in B, c \in C$ such that $c = a + b$.

3. (a) The longest path problem is to find a simple path between two vertices in a graph such that the weight of the path is maximized. Suppose we want to solve the single-source longest path problem in a directed acyclic graph. Can we modify Dijkstras algorithm to solve this problem by changing minimum to maximum? If so, then prove your algorithm correct. If not, then provide a counterexample.

   (b) In the art gallery guarding problem, we are given a line that represents a long hallway in an art gallery. We are also given a set $X = \{x_0, x_1, \ldots x_{n-1}\}$ of real numbers that specify the positions of paintings in this hallway. Suppose that a single guard can protect all the paintings within distance at most 1 of his or her position (both sides). Design an efficient algorithm for finding the placement of guards that uses the minimum number of guards to guard all the paintings with positions in X. Argue that your algorithm is correct.

4. Design an efficient algorithm to find the number of shortest paths between every pair of vertices in a weighted directed graph. What is the complexity of your algorithm ?

5. Let G be a directed graph represented using an adjacency list and each of its vertices are assigned weights $w : V \to R$. Give an efficient algorithm that computes, for every node, the highest valued node reachable from that node. What is the complexity of your algorithm ?

6. An independent set in a graph G is a subset I of the vertices such that no two vertices in I are adjacent in G. The maximum independent set problem is to find the largest independent set in G. Design an efficient algorithm to find a maximum independent set in a tree. What is the complexity of your algorithm ?

7. (a) Find $3^{216367645393787487 43} \bmod 25$

   (b) Given a positive integer n, design an efficient algorithm to decide whether there exits integers $a, b > 1$ such that $n = a^b$. What is the complexity of your algorithm ?

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.

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

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

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 oi 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

```
            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.
CS 501 Data Structures and Algorithms.
Mid Term Exam, 15 October 2015.

1. Which of the following statements are true. You must justify your answers. Answers without/wrong/enough justifications will not be given any marks here

   (a) The height of an AVL tree with n elements is $\Theta(\log n)$.(**2 marks**)

   (b) The height of a Binomial heap with n elements is $\Theta(\log n)$. (**2 marks**)

   (c) There is a linear time algorithm to determine if a string T_1 is a cyclic rotation of another string T_2. For example *algo* and *goal* are cyclic rotation of each other. (**2 marks**)

   (d) It is possible to convert a binary heap into a binary search tree in linear time. (**2 marks**)

   (e) Given an array of integers in sorted order, a balanced binary search tree can be built in linear time.(**2 marks**)

   (f) The height of a Fibonacci heap with n elements is $O(\log n)$. (**4 marks**)

2. What is the minimum and maximum number of red nodes possible in a red-black tree with 145 nodes.(**3 marks**)

3. In the dynamic table example done in the class, let us assume that the size of table is tripled each time when there is overflow (instead of doubling). The size of the table will increase as $0, 1, 3, 9, 27, 81....$ Which of the following can be used as potential function to prove that a sequence of n insertion takes $O(n)$ time in this case. (**2 marks**)

   (a) $\Phi(D) = 2$ * number of elements - size of the table.

   (b) $\Phi(D) = 3$ * number of elements - size of the table.

   (c) $\Phi(D) = 4$ * number of elements - size of the table.

   (d) $\Phi(D)$ = number of filled slots - number of empty slots.

4. You are given a array of real numbers $a_0, a_1, \ldots a_{n-1}$ and a number $k, (1 \leq k \leq n)$. You need to list the k largest elements form the given array.

   (a) Give $O(n)$ time and space algorithm.(**2 marks**)

   (b) Give $O(k)$ space and $O(n \log k)$ time algorithm.(**3 marks**)

5. Given an array of integers, we would like to compute the smallest sub-array which contains all the integers in the range $[1, \ldots k]$, where k is a constant.

   For array $A = \{1, 2, 3, 9, 12, 3, 1, 6, 3, 13, 100, 123, 4, 14, 3, 12, 23\}$ and $k = 4$, the answer is 12 as highlighted.

   Design a linear time ($O(n + k)$) algorithm for this problem.**(4 marks)**

6. You are given an array integers $A_0, A_1 \ldots A_{n-1}$, which are initially 0. We want to do the following operations (in any order) on these elements:

   - *Update*(i, x): Update the value at index i to x $(i \geq 0)$
   - *Count*(i, j): Counts the number of numbers between index i and j $(j \geq i \geq 0)$ which are divisible by 3.

   Design a data structure which uses $O(n)$ space and supports each of the above operation in $O(\log n)$ time. **(4 marks)**

International Institute of Information Technology, Bangalore
CS501 Data Structures and Algorithms.
Test 1: September 8, 2015

1. What is the complexity of the following algorithm ? (**2 marks**)

```
int Find (int n){
int i = 1, j;
while(i < n) i = i * 2;
j = n/2;
while(i > j + 1){k = (i + j)/2; if(k < n) j = k: else i = k;}
return i; }
```

2. Let $F(0) = 0, F(1) = 1, F(2) = 2$ and $F(n) = (F(n-1) + F(n-2) + F(n-3))\%1000$. We would like to have an $O(\log^2 n)$ algorithm to compute $F(n)$ by computing A^n for some matrix. What is the matrix that we should use here. (**2 marks**)

3. Let $a_1, a_2, \ldots a_n$ be a sequence of distinct numbers. The pair (i, j) is called a double inversion, if $i < j$ and $a_i > 2 * a_j$.

   (a) Give a sequence of length n, which has $\Theta(n^2)$ double inversions.(**1 marks**)

   (b) Give an $O(n \log n)$ to determine the number of double inversions in the given array. (**3 marks**)

4. Given a array of integers, we would like to compute the smallest length subarray (continues numbers from the given sequence) which contains all the integers in the range $[1, \ldots k]$, where k is a constant.

   For array $A = \{1, \mathbf{2}, \mathbf{3}, \mathbf{9}, \mathbf{12}, \mathbf{3}, \mathbf{1}, \mathbf{6}, \mathbf{3}, \mathbf{13}, \mathbf{100}, \mathbf{123}, 4, 14, 3, 12, 23\}$ and $k = 4$, the answer is 12 as highlighted.

   Design a linear time ($O(n + k)$) algorithm for this problem.(**4 marks**)

5. You are given a $n \times n$ matrix of integers such that the entries of each row are sorted (in increasing order) from left to right and the entries of each column are sorted (in increasing order) from top to bottom.

$$\begin{pmatrix} -8 & -2 & -1 & 0 & 4 \\ 3 & 8 & 15 & 25 & 26 \\ 5 & 9 & 23 & 34 & 41 \\ 14 & 18 & 33 & 49 & 51 \\ 22 & 29 & 59 & 75 & 100 \end{pmatrix}$$

(a) Given an integer x, design an $O(n)$ algorithm to find the rank of x (among the elements in the given matrix). **(2 marks)**

For example in the above matrix, rank of 24 is 12.

(b) Given x and y, with $x < y$, we would like to compute the largest **square** sub-matrix which contains numbers greater than x, but smaller than y.Give an $O(n)$ algorithm for the problem. **(3 marks)**

For example, $x = 7, y = 50$ the highlighted $3 * 3$ matrix is the answer.

$$\begin{pmatrix} -8 & -2 & -1 & 0 & 4 \\ 3 & 8 & 15 & 25 & 27 \\ 5 & 9 & 23 & 34 & 41 \\ 14 & 18 & 33 & 49 & 51 \\ 22 & 29 & 59 & 75 & 100 \end{pmatrix}$$

(c) Given x and y, with $x < y$, we would like to compute the largest **rectangular** sub-matrix which contains numbers greater than x, but smaller than y.Give an $O(n \log n)$ algorithm for the problem. **(3 marks)**

For example, $x = 7, y = 52$ the highlighted $3 * 4$ matrix is the answer.

$$\begin{pmatrix} -8 & -2 & -1 & 0 & 4 \\ 3 & 8 & 15 & 25 & 27 \\ 5 & 9 & 23 & 34 & 41 \\ 14 & 18 & 33 & 49 & 51 \\ 22 & 29 & 59 & 75 & 100 \end{pmatrix}$$

International Institute of Information Technology, Bangalore
CS 501 Data Structures and Algorithms.
Test 1: September 12, 2016

1. If $T(0) = T(1) = \Theta(1)$,then write the solution for the following recurrences. **(3 marks)**

   (a) $T(n) = 12T(n/12) + n$.

   (b) $T(n) = 9T(n/10) + 1$.

   (c) $T(n) = 7T(n/8) + n$.

2. What is the complexity of the following algorithm ? **(2 marks)**

```
int Find (int n){
int i = 1, j;
while(i < n) i = i * 2;
j = n/2;
while(i > j + 1){k = (i + j)/2; if(k < n) j = k; else i = k;}
return i; }
```

3. Give a $O(n \log k)$ time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. **(3 marks)**.

4. Given a array of n integers and an interger $1 \leq k \leq n$. We would like to find the minimum of each possible sub-arrays of size k.

   For example, $A = \{3, 9, 12, 6, 14, 5, 1, 23\}$ and $k = 4$, the minimum in each sub-array is $3, 6, 5, 1, 1$.

   Design a linear time ($O(n)$) algorithm for this problem.**(4 marks)**

5. Design an $O(n)$ time algorithm that, given a set of S of n distinct numbers and a positive number $k \leq n$, determines the k numbers in S that are closed to the median of S **(4 marks)**.

6. Given a Boolean string, the DIVIDE(i, j) function returns a k such that

   $k = i - 1$, if all the values between i and j is TRUE, else the number of TRUE values between i and k is equal to the number of FALSE values between $k + 1$ to j.

   (a) Give a linear time algorithm to implement the DIVIDE function. **(2 marks)**

   (b) UPDATE(i, b, k) returns an appropriate value of k, if the ith value is changed to b. Give an efficient algorithm to implement the UPDATE function.**(2 marks)**

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.

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

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

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)**

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
CS 501 Data Structures and Algorithms.
Practice Problems 1: August , 2014

1. Solve the following recursions (in terms of Θ). $T(0) = T(1) = \Theta(1)$ in all of the following.

   (a) $T(n) = 2T(n/2) + n$.
   (b) $T(n) = T(n/2) + n$.
   (c) $T(n) = 3T(n/2) + n$.
   (d) $T(n) = 2T(n/2) + n^2$.
   (e) $T(n) = T(n/2) + n^2$.
   (f) $T(n) = 3T(n/2) + n^2$.
   (g) $T(n) = 2T(n/2) + 1$.
   (h) $T(n) = T(n/2) + 1$.
   (i) $T(n) = 3T(n/2) + 1$.
   (j) $T(n) = 2T(n/2) + n \log n$.
   (k) $T(n) = T(n/2) + T(n/4) + 1$.
   (l) $T(n) = T(n/2) + T(n/4) + n$.
   (m) $T(n) = T(n/2) + 2T(n/4) + 1$.

2. **Maximum product Subsequence problem**

   Given an array $a_1, a_2, \ldots a_n$ of integers (both positive and negative), design a linear time algorithm to find the contiguous subsequence with the maximum product.

3. You are given an array $a_1, a_2, \ldots a_n$ of integers (both positive and negative), and a interger $l, 1 \leq n$. The length of a subsequence is defined as the number of integers in the subsequence.

   (a) Design a linear time algorithm to find the a maximum sum subsequence of lenght exactly l.
   (b) Design a linear time algorithm to find the a maximum sum subsequence of lenght at least l.

(c) Design a linear time algorithm to find the a maximum sum subsequence of lenght at most l.

4. You are given an array $a_1, a_2, \ldots a_n$ of integers (both positive and negative), and a interger l, $1 \leq n$. The density of a subsequence is defined as the sum of numbers in the subsequence devided by the length of the subsequence.

   (a) Design a linear time algorithm to find the a maximum density subsequence of lenght exactly l.

   (b) Design a linear time algorithm to find the a maximum density subsequence of lenght at most l.

   (c) Design an $O(n^2)$ algorithm to find the a maximum density subsequence of lenght at least l.

   (d) Design an $O(nl)$ algorithm to find the a maximum density subsequence of lenght at least l. **Hard**

   (e) Design a linear time algorithm to find the a maximum density subsequence of lenght at least l. **Very Hard**

5. Given two arrays A and B, containing m and n integers respectively , design an efficient algorithm to determine how many integers are in common between the two arrays.

6. The *addBlock*() operation is used to add m integers to to an existing sorted n integers, where $m << n$ (m is very small compared to n). One of your senior student gave me the following algorithm for this problem: the algorithm simply creates an array of length $n+m$, copies over the old n values into the new array, copies over the m values to the end of the array, and finally insertion sort is used (from the nth location onwards) to bring everything into order.

   (a) What is the complexity of the above algorithm.

   (b) Design an efficient algorithm for this problem.

7. Let $a_1, a_2, \ldots a_n$ be a sequence of distinct numbers. The pair (i, j) is called a inversion, if $i < j$ and $a_i > a_j$. Give an $O(n \log n)$ to determine the number of inversions in the given array.

8. Given a sorted array of distinct integers $A[0] < A[1] < \ldots A[n-1]$, design an $O(\log n)$ algorithm for the following

   (a) Decide whether there is an index i such the $A[i] = i$.

   (b) Given x and y, find the number of integers in the given array which are strictly greater than x, but strictly smaller than y.

9. Give a $O(n \log k)$ time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists.

10. Let A, B and C be three sequence of n integers each. Design an $O(n^2)$ algorithm to determine if there are three integers $a \in A, b \in B, c \in C$ such that $c = a + b$.

11. Suppose you are given two sorted arrays of integers $A[1..m]$ and $B[1..n]$ and an integer k. Describe an efficient algorithm to find the kth smallest element in the union of A and B.

    For example, given the input $A[1..10] = [0, 1, 3, 6, 11, 13, 15, 22, 32, 45]$, $B[1..5] = [2, 5, 8, 17, 29]$, $k = 9$, your algorithm should return 13. You can assume that the arrays contain no duplicates.

12. 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 $n/3 + 1$ times. Design a linear time algorithm to decided if a given sequence has a majority element.

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)**

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