# MAASAI MARA UNIVERSITY
# REGULAR
# UNIVERSITY EXAMINATIONS
# 2019/2020 ACADEMIC YEAR
# FOURTH YEAR FIRST SEMESTER

## SCHOOL OF SCIENCE AND INFORMATION SCIENCES
## DEPARTMENT OF COMPUTING AND INFORMATION SCIENCE
## BACHELOR OF SCIENCE IN INFORMATION SCIENCES

**COURSE CODE: COM 3106**
**COURSE TITLE: DESIGN & ANALYSIS OF ALGORITHM**

**DATE: 13TH DEC 2019** **TIME 14:30 – 16:30**

## INSTRUCTION TO CANDIDATE

i. Question ONE in section A is compulsory
ii. Answer any OTHER Two (2) Questions from section B
iii. Use diagrams, example and illustration where necessary
iv. All questions in section B have equal marks

**SECTION A [30 MARKS]**

**QUESTIONONE**

i. Define the term "Algorithm" **[2 Marks]**
ii. Discuss four applications of Algorithm **[8 Marks]**
iii. State any four characteristics of an Algorithm **[4 marks]**

v. Explain Algorithm creation techniques **[2 Marks]**

vi. Explain Factors for measuring good algorithm **[4 Marks]**

vii. Recall that Fibonacci numbers are defined recursively as $F_0 = 0$, $F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$. Prove the square of the n-th Fibonacci number differs from the product of the two adjacent numbers by one:

$F_n^2 = F_{n-1}.F_{n+1} + (-1)^{n+1}$ **[10 Marks]**

**SECTION B [40 MARKS]**
**QUESTION TWO [20 MARKS]**
Consider two sums, $X = x^1 + x^2 + \cdots + x^n$ and $Y = y^1 + y^2 + \cdots + y^n$ Give an algorithm that finds indices i and j such that swapping $x_i$ with $y_i$ makes the two sums equal, that is, $X - x_i + y_j = Y - y_j + x_i$, if they exist. Analyze your algorithm.

**[20 Marks]**

**QUESTION THREE [20 MARKS]**
Consider distinct items $x_1, x_2, \dots\dots, x_n$ with positive weights $w_1, w_2, \dots\dots, w_n$ such $that\ \sum_{i=1}^{n} w_i = 1.0$ . The weighted median is the item $x_k$ that satisfies

$\sum_{x_i < x_k} w_i > 0.5$, and $\sum_{x_j < x_k} w_j \leq 0.5$

(a) Show how to compute the weighted median of n items in worst-case time O (n log n) using sorting. **[10 Marks]**

(b) Show how to compute the weighted median in worst-case time O (n) using a linear-time median algorithm. **[10 Marks]**

**QUESTION FOUR [20 MARKS]**

A game-board has n columns, each consisting of a top number, the cost of visiting the column, and a bottom number, the maximum number of columns you are allowed to jump to the right. The top number can be any positive integer, while the bottom number is either 1, 2, or 3. The objective is to travel from the first column off the board, to the right of the nth column. The cost of a game is the sum of the costs of the visited columns.

Assuming the board is represented in a two dimensional array, B [2, n], the following recursive procedure computes the cost of the cheapest game:

```
int CHEAPEST(int i)
  if i > n then return 0 endif;
  x = B[1, i] + CHEAPEST(i + 1);
  y = B[1, i] + CHEAPEST(i + 2);
  z = B[1, i] + CHEAPEST(i + 3);
  case B[2, i] = 1: return x;
       B[2, i] = 2: return min{x, y};
       B[2, i] = 3: return min{x, y, z}
  endcase.
```

(a) Analyze the asymptotic running time of the procedure. **[10 Marks]**

(b) Describe and analyze a more efficient algorithm for finding the cheapest game. **[10 Marks]**

**QUESTION FIVE [20 MARKS]**

Consider a set of n intervals $[a_i, b_i]$ that cover the unit interval, that is, $[0,1]$ is contained in the union of the intervals.

(a) Describe an algorithm that computes a minimum subset of the intervals that also covers $[0,1]$. **[10 Marks]**

(b) Analyze the running time of your algorithm. **[10 Marks]**

**QUESTION SIX [20 MARKS]**

Consider a free tree and let $d(u, v)$ be the number of edges in the path connecting u to v. The diameter of the tree is the maximum $d(u, v)$ overall pairs of vertices in the tree. Design an efficient algorithm to find a spanning tree for a connected, weighted, undirected graph such that the weight of the maximum weight edge in the spanning tree is minimized. Prove the correctness of your algorithm.

(a) Give an efficient algorithm to compute the diameter of a tree. **[10 Marks]**

(b) Analyze the running time of your algorithm **[10 Marks]**

//////////////////////////////////**END**///////////////////////////