**KIBABII UNIVERSITY COLLEGE**
*(A Constituent College of MasindeMuliro University of Science Technology)*
P.O. Box 1699-50200 Bungoma, Kenya
Tel. 020-2028660/0708-085934/0734-831729
E-mail: enquiries@kibabiiuniversity.ac.ke

**UNIVERSITY REGULAR EXAMINATIONS**

**2013 /2014 ACADEMIC YEAR**

**3RD YEAR 2ND SEMESTER EXAMINATIONS**

**(MAIN EXAMINATION)**

## FOR THE DEGREE OF

## BACHELOR OF SCIENCE IN COMPUTER SCIENCE

**COURSE CODE:** CSC 124

**COURSE TITLE:** INTENET TECHNOLOGIES

**DATE:** 17TH APRIL,2014 **TIME:** 9:00A.M.-12 NOON.

---

**INSTRUCTIONS TO CANDIDATES:**

- Answer questions one and any two questions only
- Question one carries 30 marks and the other questions carry 20 marks each.

**QUESTION ONE**

(a) Give a brief history of the development of the internet. (4 marks)

(b) There are various types of connectivity to get hooked to the internet. State and explain **Four** of these facilities needed in order to get connected to the internet. (8 marks)

(c) The following are some alternatives for communication, internal and external to organizations.

(i) Internet

(ii) Intranet

(iii) Extranet

Write short notes on ALL of these headings indicating their main features and explaining how they might be used by an organization for business communication (12 marks)

(d) (i) Describe the structure of a html document (2 marks)

(ii) Describe the classification of html tags with examples. (2 marks)

(e) List the functions of the server and the client. (2 marks)

## QUESTION TWO

(a) State and explain any Five services offered on the internet. (10 marks)
(b) When developing websites, it is advisable not to incorporate too many graphics and objects. True/False? Explain. (4 marks)
(c) Explain the term protocol. (2 marks)
(d) Explain the difference between TCP/IP connection and Shell connection. (4 marks)

## QUESTION THREE

(a) Some of the most frequently used protocols include:
(i) FTP
(ii) SMTP
(iii) HTTP
Explain the uses of these protocols. (9 marks)

(b) With the growth of the internet, electronic mail services are rapidly replacing ordinary mail services. Describe the features of an E-mail service which make it more efficient than the ordinary mail service. (5 marks)
(c) Differentiate between the following: (6 marks)

(i) Web server and web browser
(ii) Website and web page
(iii) Gopher and telnet

**QUESTION FOUR**

(a) The internet technology has posed many security threats to the internet users at large. List and fully describe at least five ways in which internet privacy can be enhanced. (10 marks)

(b) The use of e-mail has become common to many internet users. However for anyone to use this facility, he or she is required to fully understand the components of an e-mail address before creating an account.

With the aid of an example, describe the FOUR components of an e-mail address (6 marks)

(c) Outline the basic steps required for a TELNET session (4 mars)

**QUESTION FIVE**

(a) After creating a web page, explain how one can use a web browser to display the contents (4marks)

(b) Write a HTML code that would display the following list in a browser window. (10 marks)

University Campuses

- Main
- Bungoma
- Nambale

Programmes on offer

1. Education
2. Science
3. Engineering

Departments

A. Computer
B. IT
C. Electrical

(c)Explain the following terms as used in Web development: (6 marks)

i. Domain name

ii. Cascading

iii. URL

iv. Scripting

v. HTML

vi. Dynamic HTML

# KIBABII UNIVERSITY COLLEGE

# THIRD YEAR EXAMINATION

# FOR THE DEGREE OF

# BACHELOR OF SCIENCE IN COMPUTER SCIENCE

# MARKING SCHEME

COURSE CODE & TITLE: CSC 311: DESIGN AND ANALYSIS OF ALGORITHMS

## QUESTION ONE

(a) What is meant by computational complexity and why is it important? Discuss the relative importance of processor speed vs algorithm complexity for practical computing. (3 marks)

**Soln: complexity of an algorithm is a measure of its efficiency usually expressed in terms of the amount of computation efforts (in time or storage) required to process input of a given size. Both processor speed and algorithm complexity are important for practical computing but for a highly inefficient algorithm even large increases in processor speed will make little impact on the time required.**

(b) Often a given problem can be solved using different techniques but one technique results in a much faster algorithm than the other. Explain what is meant by an algorithm and technique. (3 marks)

**Soln: an algorithm is a problem solving rule while a technique is the approach or methodology used to solve a problem.**

(c) Briefly describe any FOUR algorithmic techniques giving an example of a problem solution that applies the technique (8 marks)

**Soln: Any of the following well explained to earn 2 marks each**

- **Brute force**
- **Divide – and – conquer**
- **Dynamic programming.**
- **Greedy algorithms**
- **Kruskal's**
- **Dijkstra etc.**

(d) Often one can check whether all elements of an array are distinct by a two-part algorithm based on the array's presorting. Give the algorithm for checking element uniqueness for the sorted array and determine its efficiency. (6 marks)

**Soln:**

```
Algorithm: uniqueElement A[0 ----- n-1]
// checks whether all elements in a given array are distinct
//Inputs: An array A[0 ----- n-1]
//Outputs: Returns "true" if all elements in A are distinct and "false" Otherwise.
        for i<-0 to n-2 do
        for j<-i+1 to n-1 do
        if A[I] = A[j] return false
return true
```

(e) (i) Explain the two kinds of efficiencies. (4 marks)

**Soln: Time efficiency: - indicates how fast the algorithm runs. Space efficiency: - indicates how much extra memory the algorithm needs.**

(ii) Explain the Worst case and Best case efficiencies analysis (3 marks)

**Soln: worst case efficiency of an algorithm is its efficiency for the worst case inputs of size n which is an input of size n for which the algorithm runs the longest among all possible inputs of that size. Best case efficiency is the efficiency of the best case input of size n which is an input of size n for which the algorithm runs the fastest among all possible inputs of that size.**

Consider the following concepts that are among a sequence of steps one typically goes through in designing and analyzing an algorithm. For each of them explain briefly what it means and how it applies to the design and analysis of algorithms. (4 marks)

(i) Understand the problem.

**Soln: The first thing you need to do before designing the algorithm is to understand completely the problem given. If the problem in question is one of the types that arise in computing applications quite often, you might be able to use a known algorithm for solving it. A correct algorithm in not one that works most of the time but one that works correctly for all legitimate inputs**

(ii) Choosing between Exact and approximate algorithm.

**Soln: Another principle decision is to choose between solving the problem exactly or solving it approx. The former is called an exact algorithm and the later is an approximate algorithm. One may opt for the approx. for those problems which cannot just be solved exactly. Secondly available algorithms for solving a problem exactly may be an acceptably slow because of the problem's intrinsic complexity.**

(iii) Proving and algorithm's correctness.

**Soln: One an algorithm has been specified, you have to prove its correctness. i.e. you have to prove that the algorithm yields a required result for every legitimate input in a finite amount of time. A common technique for proving correctness is to use mathematical induction because an algorithm's iterations provide a natural sequence of steps needed for such proofs.**

(iv) Analyzing an algorithm.

**Soln: after correctness, by far the most important quality we want how algorithm to posses its efficiency.**

## QUESTION TWO

(a) When analyzing the efficiency of algorithms, we talk about the algorithm's basic operation. What do you understand by the term "basic operation"? (3 marks)

**Sol: The operation contributing the most to the total running time and computes the no. of times the basic operation is executed.**

(b) Let $c_{op}$ be the time of execution of an algorithm's basic operation on a particular computer and let *C(n)* be the number of times this operation needs to be executed for this algorithm. Give a formula that estimates the running time *T(n)* of a program implementing this algorithm. (1 mark)

**Sol: T(n) = Cop C(n)**

(c) Assuming $C(n) = \frac{1}{2} n(n-1)$ in part (b) above and n is large, how much longer will the algorithm run if we double the size of the input? Show how you arrive at your answer.

**Sol:**

$$C(n) = \frac{1}{2} n(n-1)$$

$$\text{doubling } c(2n) = \frac{1}{2} (2n)\,(2n-2) = (n(2n-2)$$

$$\frac{n(2n-2)}{\frac{1}{2} n(n-1)} = \frac{2n(2n-2)}{n(n-1)} = 4$$

(d) The efficiency analysis framework concentrates on the order of growth of an algorithm's basic operation count as the principal indicator of the algorithm's efficiency. To compare and rank such order of growth we use the three notations: O (big oh), Ω (big omega) and θ (big theta). Explain these notations. (6 marks)

**Sol:**

**O (big oh): set of all functions with the same or smaller order of growth as g(n) as n grows to infinite, Ω (big omega): set of all functions with larger or same order of growth as g(n) as n grows to infinite, and θ (big theta): set of all functions with same order of growth as g(n) as n grows to infinite.**

(e) Give the limit based expression for comparing the orders of growth of two specific functions t(n) and g(n) explaining the three principal cases that may arise. (4 marks)

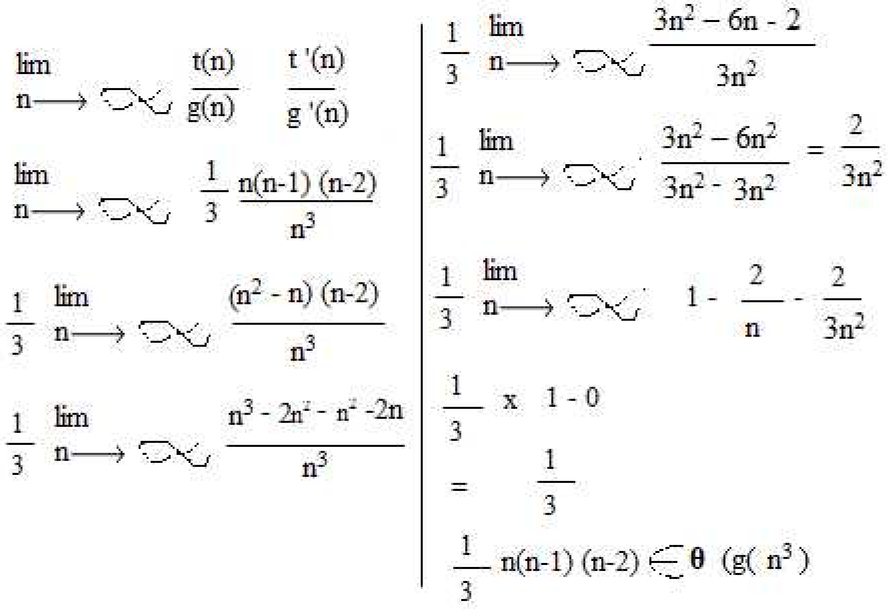
**Sol:**

$$\lim_{n \to \infty} \frac{t(n)}{g(n)} \begin{cases} 0 & t(n) \text{ has smaller order of growth than } g(n) \\ c & t(n) \text{ has same order of growth as } g(n) \\ \infty & \text{has larger order of growth than } g(n) \end{cases}$$

(f) Use the limit based approach to compare orders of growth *of* $\frac{1}{3}n(n-1)(n-2)$ and $n^3$.

Using an appropriate notation indicate their relative orders of growth. (6 marks)

**Sol:**



$$\lim_{n\to\infty} \frac{t(n)}{g(n)} \quad \frac{t'(n)}{g'(n)}$$

$$\lim_{n\to\infty} \frac{1}{3}\,\frac{n(n-1)\,(n-2)}{n^3}$$

$$\frac{1}{3}\lim_{n\to\infty} \frac{(n^2-n)\,(n-2)}{n^3}$$

$$\frac{1}{3}\lim_{n\to\infty} \frac{n^3-2n^2-n^2-2n}{n^3}$$

$$\frac{1}{3}\lim_{n\to\infty} \frac{3n^2-6n-2}{3n^2}$$

$$\frac{1}{3}\lim_{n\to\infty} \frac{3n^2-6n^2}{3n^2-3n^2} = \frac{2}{3n^2}$$

$$\frac{1}{3}\lim_{n\to\infty} \quad 1-\frac{2}{n}-\frac{2}{3n^2}$$

$$\frac{1}{3} \times 1-0$$

$$= \frac{1}{3}$$

$$\frac{1}{3}\,n(n-1)\,(n-2) \in \boldsymbol{\theta}\;(g(\,n^3\,)$$

## QUESTION THREE

(a) With respect to graphs explain the following terms. (4 marks)

(i) Directed graph

**Sol: a graph in which the edge (u,v) is directed form vertex u to v and the graph itself is called directed or digraph (easy to trace a path from a vertex to the other)**

(ii) Complete graph

**Sol: a graph with every pair of its vertices connected by an edge.**

(iii) Sparse graph

**Sol: a graph with few edges relative to the no. of its vertices.**

(iv) Acyclic graph

**Sol: a graph with no cirles.**

(b) (i) Describe two principal methods for representing graphs for computer algorithms (4 marks)

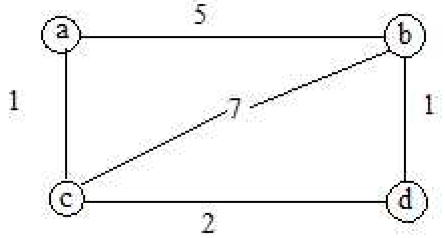
**Sol: adjacency matrix:- adjacency matrix of undirected graph is always symmetric. ie A[i,j] = A[j,i]**

**adjacency linked list:- of a graph of a digraph is a collection of linked lists one for each vertex that contains all the vertices adjacent the list's vertices.**

(ii) If a graph is sparse which representation will you use and why? (2 marks)

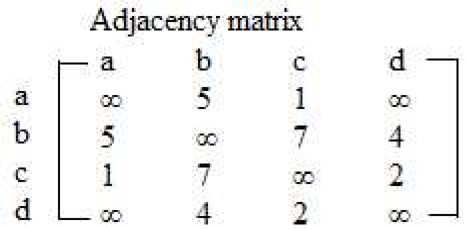
**Sol: the adjacency linked list representation may use less space than the corresponding adjacent matrix.**

(c) Consider the weighted graph given below;

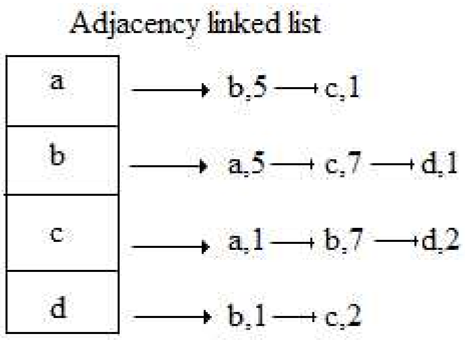


Represent the weighted graph using the two representation methods described in part (i) of (c) above (4 marks)

**Sols:**


Adjacency matrix

$$\begin{array}{c|cccc} & a & b & c & d \\ a & \infty & 5 & 1 & \infty \\ b & 5 & \infty & 7 & 4 \\ c & 1 & 7 & \infty & 2 \\ d & \infty & 4 & 2 & \infty \end{array}$$



(d) Briefly describe any Three problem types (6 marks)

**Sol: Any three of the following well explained:**

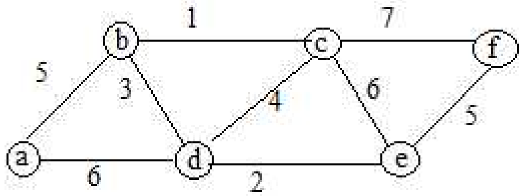
- **Sorting**
- **Searching**
- **String processing**
- **Graph**
- **Combinatorial**
- **Numerical**
- **Geometric**

## QUESTION FOUR

(a) Dijkstra algorithm is an example of the Greedy algorithm approach. True or False? Give reasons. (3 marks)

**Sol: True: because at every stage of the algorithm the local optimal choice is taken without considering whether this will lead to a globally optimal solution or not.**

(b) Using Dijkstra's algorithm, determine the single-source shortest path from vertex 'a' to all the other vertices. (7 marks)



**Sol:**

**a(-,0) b(a,5) c(-,∞) d(a,6) e(-,∞) f(-,∞)**

**b(a,5) c(b,6) d(a,6) e(-,∞) f(-,∞)**

**c(b,6) d(a,6) e(c,12) f(c,13)**

**d(a,6) e(d,8) f(c,13)**

**e(d,8) f(c,13)**

**f(c13)**

| | | |
|---|---|---|
| **a to b** | **a ->b** | **5** |
| **a to c** | **a->b->c** | **6** |
| **a to d** | **a->d** | **6** |
| **a to e** | **a->d->e** | **8** |
| **a to f** | **a->b->c->f** | **13** |

(c) Give a brief description of each of the following algorithmic design techniques. (6 marks)

(i) Backtracking

(ii) Branch-and-bound

**Sol:**

**Both :**

**- make it possible to solve some large instance of combinatorial problem.**

- **Improvement of exhaustive search**
- **Construct candidate solution one component at a time and evaluate the partially constructed solution.**
- **Terminate a node as soon as it can be guaranteed that no solution to the problem can be obtained by considering choices that correspond to the nodes descendants.**

**B/B: Applicable to both optimization problems and non optimization as well.**

**B/T: Applicable on non optimization.**

**B/T: State space tree is generated depth first**

**B/T: If no legitimate option for the next component is available, the algorithm backtracks to the previous node.**

(iii) What does dynamic programming has in common with divide-and-conquer? (2marks)

**Sol: Both techniques divide the problem into smaller sub problems**

(iv)What is the principal difference between the two techniques? (2 marks)

**Sol: D/P is a bottom up technique where the results of the sub problems are re-used in solving bigger sub problems. Whereas D/C is a top down technique.**

## QUESTION FIVE

(a) Define the following terms;

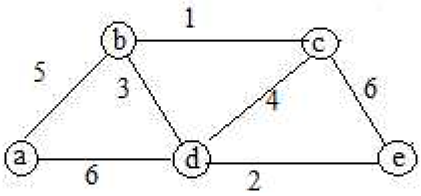
(i) Spanning tree

**Sol: is a connected a cyclic sub graph of a connected graph ie . a tree that contains all the vertices of the graph**

(ii) Minimum spanning tree (2 marks)

**Soln: of a weighted connected graph is its spanning tree of the smallest weight.**

(b) Apply Prim's or Kruskal's algorithms to find a minimum spanning tree of the following graph. (7 marks)



**Soln:**

**a(-,0) b(a,5) c(a,7) d(-∞) e(a,2) f(-,∞)**

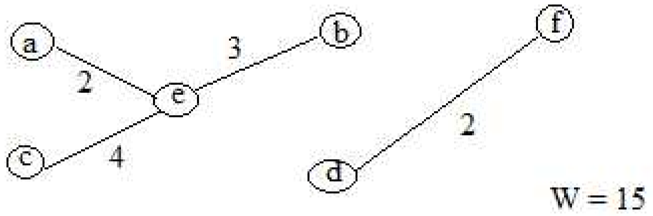
**e(a,2) b(e,3) c(e,4) d(e,5) f(-,∞)**

**b(e,3) c(e,4) d(e,5) f(b,8)**

**c(e,4) d(c,4) f(b,8)**

**d(c,4) f(d,2)**

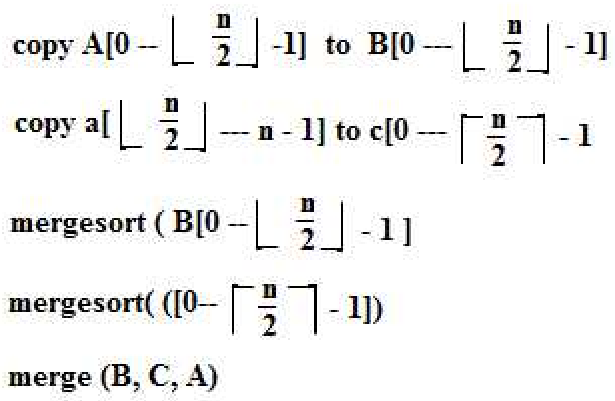
**f(d,2)**



(c) Describe the recursive merge-sort algorithm, illustrating how it works when applied to the list [7,3,4,1,2,6,8,5] (6 marks)

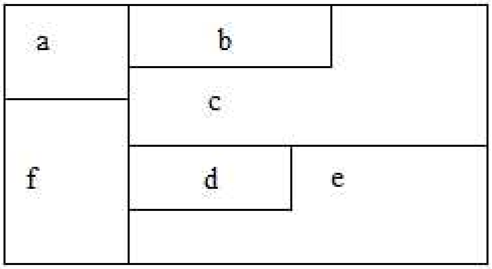
**Soln:**

```
Algorithm Mergesort (A[0 ----- n-1])
//sorts array A[0 --- n-1] by recursive mergesort
Input: An array A[0 – n-1] of ordered elements
//output: An array A[0 – n-1] sorted in non decreasing order
If n>1
Copy A[0 –LN/2
```



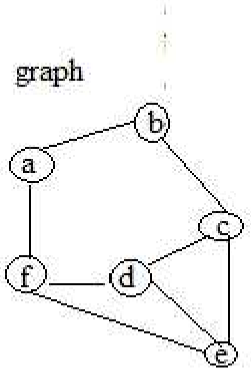
```
copy A[0 -- $\lfloor \frac{n}{2} \rfloor$ -1]  to  B[0 --- $\lfloor \frac{n}{2} \rfloor$ - 1]
copy a[$\lfloor \frac{n}{2} \rfloor$ --- n - 1] to c[0 --- $\lceil \frac{n}{2} \rceil$ - 1
mergesort ( B[0 -- $\lfloor \frac{n}{2} \rfloor$ - 1 ]
mergesort( ([0-- $\lceil \frac{n}{2} \rceil$ - 1])
merge (B, C, A)
```

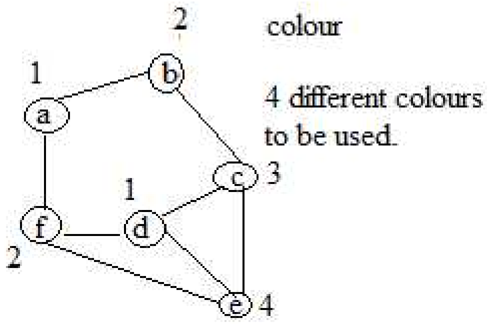
(d) Consider the following map;



Model the map as a graph and hence color the map with the smallest number of colors so that no two neighboring regions have the same color. (5 marks)

**Soln:**


graph
a
b
c
f
d
e


2
colour
1
a
b
4 different colours
to be used.
c
3
1
f
d
2
e
4