QUIZ MARKING GUIDE

1. Explain the properties of an algorithm

SOLUTION

Algorithms generally share a set of properties:

i. **Input:** what the algorithm takes in as input
ii. **Output**: what the algorithm produces as output
iii. **Definiteness:** the steps are defined precisely
iv. **Correctness:** should produce the correct output
v. **Finiteness:** the steps required should be finite
vi. **Effectiveness**: each step must be able to be performed in a finite amount of time
vii. **Generality:** the algorithm should be applicable to all problems of a similar form

2. Distinguish between Pseudocode and Algorithm

SOLUTION

Parameter	Algorithm	Pseudocode
Definition	A sequential set of orders to complete certain task in a program	A "text-based" tool useful in developing algorithm
Aim	To help in performing the task and get the desired output through defined steps	To simplify the programming language so that humans can understand without having prior knowledge about programming language
Characteristics	Clear beginning and end, usage of named variables and identifiers	Clear, unambiguous, defined input and output, language-independent and feasible

Parameter	Algorithm	Pseudocode
Advantages	Step-wise representation which is simple and easy to understand and executes on available resources	Use of simple English language, designs the entire flow of the program, and can be easily converted to actual programming code
Disadvantages	Time-consuming and certain branch and loop statements are difficult to depict in algorithm	It cannot be compiled or executed and every designer has a different style of writing pseudocode

3. Explain the best case, worst case and average case for algorithm analysis

SOLUTION

Generally, we perform the following types of algorithm analysis:

- **Worst-case:** The maximum number of steps taken on any instance of size a.
- **Best-case:** The minimum number of steps taken on any instance of size a.
- **Average case**: An average number of steps taken on any instance of size a.

Worst-case complexity:
Maximum time required for program execution (Run slowest among all inputs) In the worst-case analysis, we calculate upper bound on running time of an algorithm.

Average Complexity:
Average time required for program execution. Gives the necessary information about algorithm's behavior on random input

Best Case Complexity:
Minimum time required for program execution (Run Fastest among all inputs). It gives lower bound on running time of algorithm for any instances of input.

4. Give the conditions for satisfying the asymptotic notations

SOLUTION

1. Big O Notation

f(n)=O(g(n)) (read: f of n is big oh of g of n), if there exists a positive integer **n0** and a positive number **c** such that $|\mathbf{f(n)}| \leq \mathbf{c}|\mathbf{g(n)}|$, for all $\mathbf{n} \geq \mathbf{n0}$.

2. Omega (Ω) Notation

f(n)=Ω(g(n)) (read: f of n is omega of g of n), if there exists a positive integer **n0** and a positive integer **c** such that $|\mathbf{f(n)}| \geq \mathbf{c}|\mathbf{g(n)}|$, for all $\mathbf{n} \geq \mathbf{n0}$.

3. Theta (Θ) Notation

f(n)= Θ(g(n)) (read: f of n is thita of g of n), if there exists a positive integer **n0** and a positive integer **c1** and **c2** such that $\mathbf{c1}|\mathbf{g(n)}| \leq |\mathbf{f(n)}| \leq \mathbf{c2}|\mathbf{g(n)}|$, for all $\mathbf{n} \geq \mathbf{n0}$.

5. Give the meaning of the following asymptotic notations

SOLUTION

i.

Running time T(n)	Complexity O(n)
$n^2 + 100n + 1$	$O(n^2)$
$0.001n^3 + n^2 + 1$	$O(n^3)$
$23n$	$O(n)$
2^{3n}	$O(8^n)$ as $2^{3n} \equiv (2^3)^n$
2^{3+n}	$O(2^n)$ as $2^{3+n} \equiv 2^3 \cdot 2^n$
$2 \cdot 3^n$	$O(3^n)$

ii

Running time T(n)	Complexity O(n)
$0.0001n + 10000$	$O(n)$
$100000n + 10000$	$O(n)$
$0.0001n^2 + 10000n$	$O(n^2)$
$100000n^2 + 10000n$	$O(n^2)$
$30 \log_{20}(23n)$	$O(\log n)$ as $\log_c(ab) = \log_c a + \log_c b$