www.csepedia.com

Punjab Technical University

B.Tech. – Computer Science & Engineering (Sem. – 5th)

Design and Analysis of Algorithms

Subject Code: CS-307

Q:-1) Define Recurrence Relation? (May 2013)

Ans:-1) **Recurrence Relation**

Consider a sequence $\{a_n\}$. A **recurrence relation** for the sequence is an equation that relates the general term a_n with some of its preceding terms a_0, a_1 … a_n-1.

Q:-2) What do you mean by Worst Case Analysis? (May 2013)

Ans:-2) **Worst Case Analysis**

In the worst case analysis, we calculate upper bound on running time of an algorithm. We must know the case that causes maximum number of operations to be executed. For Linear Search, the worst case happens when the element to be searched (x in the above code) is not present in the array. When x is not present, the search() functions compares it with all the elements of arr[] one by one. Therefore, the worst case time complexity of linear search would be O(n).

Q:-3) Define Non-Deterministic Algorithm. (May 2013)

Ans:-3) **Nondeterministic Algorithm:**

A nondeterministic algorithm is one in which for a given input instance each intermediate step has one or more possibilities. This means that there may be more than one path from which the algorithm may arbitrarily choose one. Not all paths terminate successfully to give the desired output. The nondeterministic algorithm works in such a way so as to always choose a path that terminates successfully, thus always giving the correct result.

Procedures of a Nondeterministic Algorithm:

The nondeterministic algorithm uses three basic procedures as follows:

1. **CHOICE (1, *n*) or CHOICE(*S*):** This procedure chooses and returns an arbitrary element, in favor of the algorithm, from the closed interval [1,*n*] or from the set *S*.
2. **SUCCESS:** This procedure declares a successful completion of the algorithm.
3. **FAILURE:** This procedure declares an unsuccessful termination of the algorithm.

Q:-4) Give brief concept of Divide and Conquer. (May 2013)

Ans:-4) **Divide-and-conquer**

The Divide and Conquer strategy can be viewed as one which has three steps.

The first step is called **Divide** which is nothing but dividing the given problems into smaller sub problems which are identical to the original problem and also these sub problems are of the same size.

The second step is called **Conquer** where in we solve these sub problems recursively.

The third step is called **Combine** where in we combine the solutions of the sub problems to get the solution for the original problem.

Q:-5) What is Stable Sorting. (May 2013)

Ans:-5) A sorting algorithm is said to be stable if two objects with equal keys appear in the same order in sorted output as they appear in the input unsorted array. Some sorting algorithms are stable by nature like Insertion sort, Merge Sort, Bubble Sort, etc. And some sorting algorithms are not, like Heap Sort, Quick Sort, etc.

Q:-6) What is the various steps in the design of an algorithm. (May 2013)

Ans:-6) Algorithm design is a specific method to create a mathematical process in solving problems. Algorithm design is identified and incorporated into many solution theories of operation research, such as dynamic programming and divide-and-conquer.

One of the most important aspects of algorithm design is creating an algorithm that has an efficient run time, also known as its big Oh.

Steps in development of Algorithms

1. Problem definition
2. Development of a modal
3. Specification of Algorithm
4. Designing an Algorithm
5. Checking the correctness of Algorithm
6. Analysis of Algorithm
7. Implementation of Algorithm
8. Program testing
9. Documentation Preparation

Q:-7) What is NP Hard Problem. (May 2013)

Ans:-7) **What is NP?**

NP is the set of all decision problems (question with yes-or-no answer) for which the 'yes'-answers can be ***verified*** in polynomial time $O(n^k)$ where n is

the problem size, and k is a constant) by a deterministic Turing machine. Polynomial time is sometimes used as the definition of *fast* or *quickly*.

What is P?

P is the set of all decision problems which can be **solved** in polynomial time by a deterministic Turing machine. Since it can solve in polynomial time, it can also be verified in polynomial time. Therefore P is a subset of NP.

What is NP-Complete?

A problem x that is in NP is also in NP-Complete if and only if every other problem in NP can be quickly (ie. in polynomial time) transformed into x. In other words:

1. x is in NP, and
2. Every problem in NP is reducible to x

So what makes NP-Complete so interesting is that if any one of the NP-Complete problems was to be solved quickly then all NP problems can be solved quickly.

What is NP-Hard?

NP-Hard are problems that are at least as hard as the hardest problems in NP. Note that NP-Complete problems are also NP-hard. However not all NP-hard problems are NP (or even a decision problem), despite having 'NP' as a

prefix. That is the NP in NP-hard does not mean 'non-deterministic polynomial time'. Yes this is confusing but its usage is entrenched and unlikely to change.

Q:-8) What are the conditions under which Backtracking can be used. (May 2013)

Ans:-8) **Backtracking** is a solution strategy that may be implemented using recursion and a sequence of guesses that ultimately lead to a solution.

If a particular guess leads to an impasse (no solution), we retrace our steps in reverse order to replace our last guess with another option, try to complete the solution again.

Q:-9) Define Time and Space complexity. (Dec. 2011)

Ans:-9) *Efficiency* of an algorithm can be measured in terms of:

***Time complexity*:** A measure of the amount of time required to execute an *algorithm.*

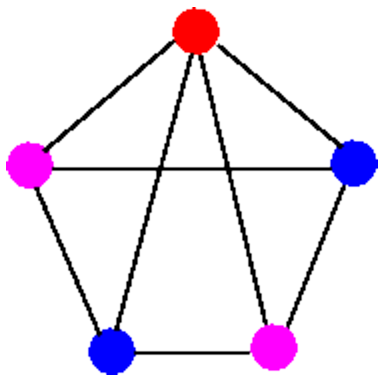
***Space complexity*:** The amount of memory required.

Q:-10) What is String Matching Algorithm. (Dec. 2011)

Ans:-10) **String matching** is fundamental to database and text processing applications. Every text editor must contain a mechanism to search the current document for arbitrary strings. Pattern matching programming languages such as Perl derive much of their power from their built-in string matching primitives, making it easy to fashion programs that filter and modify text. Spelling checkers scan an input text for words in the dictionary and reject any strings that do not match.

Q:-11) List the uses of Graph coloring. (Dec. 2011)

Ans:-11) The graph (or vertex) coloring problem, which involves assigning colors to vertices in a graph such that adjacent vertices have distinct colors, arises in a number of scientific and engineering applications such as **scheduling , register allocation , optimization and parallel numerical computation.**



The least number of colors require to color the vertices of a graph so that the adjacent vertices do not have the same color is called as the chromatic number.

In above Graph the Chromatic number is 3 (No. of Colors)

Q:-12) What do you mean by Dynamic Programming. (Dec. 2011)

Ans:-12) **Dynamic programming is an optimization technique.**

Greedy vs. Dynamic Programming:

- Both techniques are optimization techniques, and both build solutions from a collection of choices of individual elements.
- The greedy method computes its solution by making its choices in a serial forward fashion, never looking back or revising previous choices.

- Dynamic programming computes its solution bottom up by synthesizing them from smaller sub solutions, and by trying many possibilities and choices before it arrives at the optimal set of choices.

Divide and Conquer vs. Dynamic Programming:

- Both techniques split their input into parts, find sub solutions to the parts, and synthesize larger solutions from smaller ones.
- Divide and Conquer splits its input at pre specified deterministic points (e.g., always in the middle)
- Dynamic Programming splits its input at every possible split points rather than at pre-specified points. After trying all split points, it determines which split point is optimal.

Q:-13) Write the Worst Case running time of Merge Sort. (Dec. 2011)

Ans:-13) **Complexity of Merge Sort**

Worst case O (n log n)

Q:-14) State Principle of Optimality. (Dec. 2012)

Ans:-14) Principle of Optimality

A problem is said to satisfy the Principle of Optimality if the sub-solutions of an optimal solution of the problem are themselves optimal solutions for their sub-problems.

Q:-15) what do you mean by Control Abstraction? (Dec. 2012)

Ans:-15) In computer programming, abstraction can apply to control or to data: **Control abstraction** is the abstraction of actions while **data abstraction** is that of data structures.

- Control abstraction involves the use of subprograms and related concepts control flows
- Data abstraction allows handling data bits in meaningful ways.

Q:-16) Define the Terms: E-Node, Live Node, Dead Node (Dec. 2012)

Ans:-16) **Live node** is a node that has been generated but whose children have not yet been generated.

E-node is a live node whose children are currently being explored. In other words, an E-node is a node currently being expanded.

Dead node is a generated node that is not to be expanded or explored any further.

Q:-17) What is the Time complexity of Selection Sort. (May 2010)

Ans:-17) **Time complexity of Selection Sort**

$O(n^2)$

Q:-18) Define Asymptotic Notations. (May 2010)

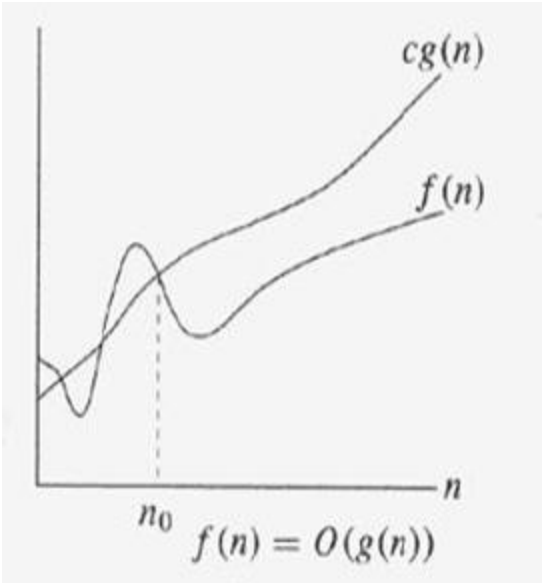
Ans:-18) **Big Oh Notation**

$f(n) = O(g(n))$

if there are positive constants c and nosuch that $f(n) \leq cg(n)$ for all $n \geq n_0$ and c>0 This notation is known as Big-Oh notation.

Example:-

$n^2 + 3n + 4 \leq 2n^2$ *for all* $n_0 > 10$ *is* $\boldsymbol{O(n^2)}$

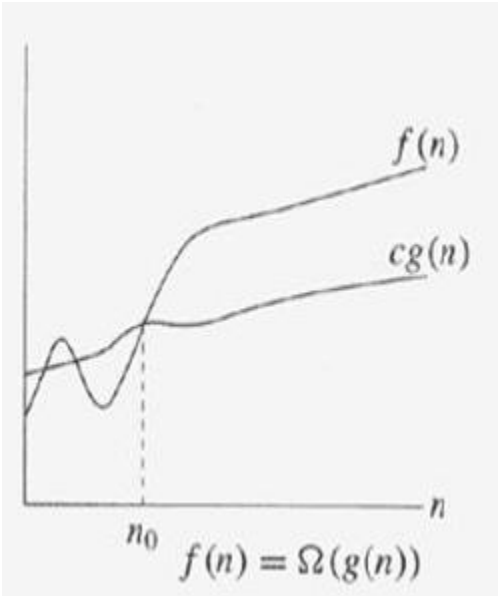


Big Omega Notation

f(n)=Ω(g(n))

if there are positive constants c and no such that f(n)>=c g(n) for all n>=n_o.

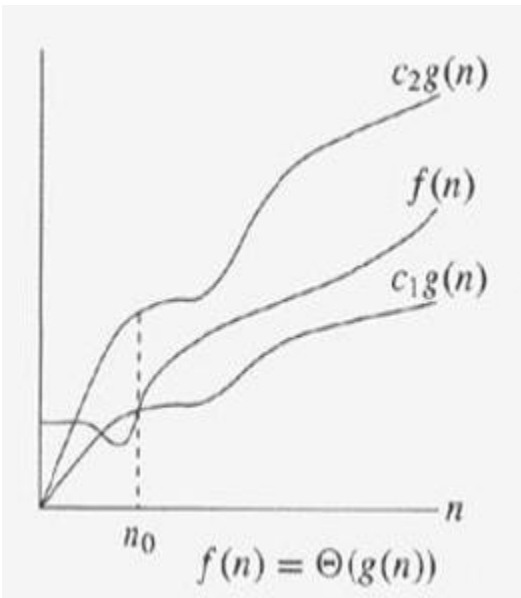
This notation is known as Big-Omega notation.



Big Theta Notation

$f(n) = \Theta(g(n))$

If there are positive constants c1, c2 and no such that $c_1g(n) <= f(n) <= c_2g(n)$, for all $n >= n_0$. This notation is known as Big-Theta notation.



Q:-19) What is LC Search? (May 2011)

Ans:-19) **LC Search Method**

A search strategy that choose the E-Node that costs less in the next step--- that is we always choose the node that can be reached with minimal cost from the current location.

What is Greedy Method?

Optimization problems are problems where in we would like to find the best of all possible solutions. In other words we need to find the solution which has the optimal (maximum or minimum) value satisfying the given constraints.

In the greedy approach each step chosen has to satisfy the constraints given in the problem. Each step is chosen such that it is the best alternative among all feasible choices that are available. The choice of a step once made cannot be changed in subsequent steps.

What is Branch and Bound?

This technique refers to all state space search methods in which all the children of a node are generated before any other node can become the live node. The search for a new node can not begin until the current node is not fully explored, As in case of backtracking, bounding functions are used to avoid the generation of sub trees that don't contain an answer node.

What is Order of Complexity?

Generally, an algorithm has an asymptotic computational complexity. Assuming the input is of size N, we can say that the algorithm will finish at $O(N)$, $O(N^2)$, $O(N^3)$, $O(N*\log(N))$ etc. This means that it is a certain

mathematical expression of the size of the input, and the algorithm finishes between two factors of it.

Generally, the smaller the order of complexity of the program's underlying algorithm, the faster it will run and the better it will scale as the input gets larger. Thus, we should often seek more efficient algorithms in order to reduce the order of complexity.