**Sarvodaya College**

**(AFFILIATEDTOSAURASHTRAUNIVERSITY)**

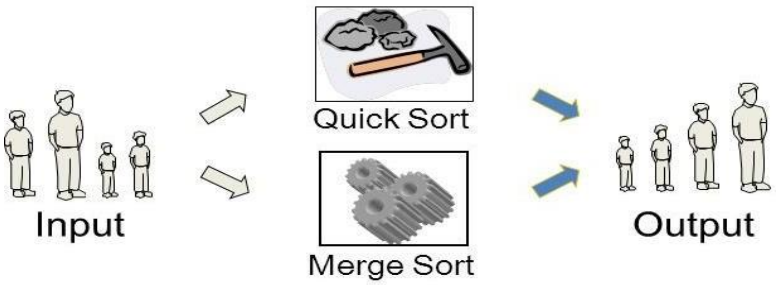
# UNIT -1–PART–1-ALGORITHMANALYSIS

- Theanalysisofalgorithm.
- Timeandspacecomplexities.
- Asymptoticnotation.
- Classesofalgorithm.
- Big-OhNotation
- Big-OmegaNotation

## 1. Theanalysisofalgorithm

An **algorithm** is a step-by-step procedure for solving a problem in a finite amount of time.

How to evaluate algorithms?




- Which one is better?
- What are the criteria?

- An algorithm is a step by step sequence of instruction to solve thecomputational problem in a finite amount of time in an Englishlanguage.
- An algorithm can be written in English but we are interested inalgorithms which have been precisely specified using an appropriatemathematicalformalism—suchasprogramming language.
- Everyalgorithmshouldhavethefollowingfivecharacteristics:
  - **i.** Input---Thealgorithmshouldtakezeroormoreinput.
  - **ii.** Output---Thealgorithmshouldproduceoneormoreoutputs.

iii. Definiteness---Each and every step of algorithm should bedefinedunambiguously.

iv. Effectiveness---Ahumanshouldbeabletocalculatethevaluesinvolvedintheprocedureofthealgorithmusingpaperanpencil.

v. Termination---An algorithm must terminated after a finitenumberofsteps.

- **Complexityofanalgorithm**isthemeasureofanalysisofalgorithm.Itis alsoknownascomputationalcomplexity.
- Analyzing an algorithm means predicting the resources that thealgorithm requires such as memory, communication, bandwidth,logicgatesantime.
- Theanalysisoftheprogramrequirestwomainconsiderations:

i. **SpaceComplexity**

ii. **TimeComplexity**

**2. Timeandspace Complexities.**

- **Time**
  - **Executing instructions take time**
  - **How fast does the algorithm run?**
  - **What affects its runtime?**
- **Space**
  - **Data structures take space**
  - **What kind of data structures can be used?**
  - **How does the choice of data structure affect the runtime?**

- Thetimecomplexity of aprogram/algorithm istheamountofcomputertimethat itneedsto runto completion.
- The space complexity of a program/algorithm is the amount ofmemorythatitneedsto runto completion.

i. **SPACECOMPLEXITY**

- Theamountofmemoryrequiredtorunandcompletionofa nalgorithmorprogramisknownasspacecomplexity.

- Its analysis is known as analysis of space complexityofanalgorithmor program.
- Therearespecificreasonsavailableforstudyingspaceco mplexity:
  - ✓ If the program is run on multi user system thenitmay berequired to specify theamount ofmemoryto be allocatedto the program.
  - ✓ Toknowinadvancethatsufficientmemoryisavaila bleornot to runtheprogram.
  - ✓ Theremaybeseveralpossiblesolutionswithdiffere nt spacerequirements.
  - ✓ Itcanbeusedtoestimatethesizeofthelargestproble mthatprogramcansolve.
- Thespaceneededbyaprogramconsistsoffollowingcomp onents.
  - ✓ Instructionspace
  - ✓ Dataspace
  - ✓ Environmentstackspace

## ii. TIMECOMPLEXITY

- The time complexity of an algorithm or a program istheamountoftimeitneeds torumtocompletion.
- The exact time will depend on the implementation ofthealgorithm,programminglanguage,optimizingcapa bilitiesofthecompilerusedandso on…
- Someofthereasons forstudyingtime complexityare:
  - ✓ Wemaybeinterestedtoknowinadvancewhether the program will provide a satisfactoryrealtimeresponse.
  - ✓ There may be several possible solutions withdifferent timerequirement.
- When we analyze an algorithm depends on the inputdata,therearethreedifferenttypesoftimecomplexiti eswhichcanbeanalyzed foranalgorithm.
  - ✓ Bestcasetimecomplexity
  - ✓ Averagecasetimecomplexity
  - ✓ Worstcasetimecomplexity

## 3. Asymptotic notation

- We are usually interesting in the order of growth of the running time of an algorithm, not in the exact running time. This is also referredto astheasymptoticnotation.
- We need to develop a way to talk about rate of growth of functionsso that we cancompare algorithms.
- Asymptotic notations are mathematical tools to represent time

compexity of algorithms for Asymptotic notation.

The main idea of asymptotic analysis is to have a measure of efficiency of algorithms that doesn't depend on machine specific constants, and doesn't require algorithms to be implemented and time taken by programs to be compared.

## 4. Classesofalgorithm

- **Byimplementationway**
  - **i.** Itisfurtherclassified into followingsubcategories.
    - RecursionorIteration
      - ✓ Arecursivealgorithmmeansthatitinvokesitselfrepeatedlyuntilacertainconditionmatches.
      - ✓ An iterative algorithms use repetitive constructslikeloopsandsometimesadditionaldata structureslikestackstosolvethegivenproblems.
    - Logical
      - ✓ Thelogiccomponentexpressestheaxioms(maximum)thatmaybeusedinthecomputationand the control component determines the wayinwhichdeductionisapplied tothe axioms.
    - SerialorParallelorDistributed
      - ✓ Acomputerwhichcanexecute oninstructionofanalgorithmatatimeisknownasserialcomputers. An algorithm designed for such anenvironmentis calleda serialalgorithm.
    - DeterministicofNon-Deterministic
      - ✓ Deterministicalgorithmssolvetheproblemwithexactdecisionateverystep ofthe algorithm.

- ExactorApproximate
  - ✓ While many algorithms reach an exact solution,approximationalgorithmstrytofindanapproximationthatisclosetothetruesolution.
- QuantumAlgorithm
  - ✓ Thisrunsonarealisticmodelofquantumcomputation.
  - ✓ These algorithms use some essential feature ofquantumcomputationsuchasquantumsuperpositionorquantumembarrassingsituation.

- **Bydesignparadigm**
  - **i.** There is a number of paradigms which is different from eachother.Also,itwillincludemanydifferenttypesofalgorithm.
    - Brute-forceorexhaustivesearch
      - ✓ Thisisthenaturalmethodoftryingeverypossible solutiontoseewhich isbest.
    - Divideandconquer
      - ✓ Adivideandconqueralgorithmrepeatedlyreduces into smaller problems until the problemsarenot enoughto solveeasily.
    - Dynamicprogramming
      - ✓ When a problem shows optimal substructure,meaning the optimal solution to a problem canbe constructed form optimal solutions to subproblems,andoverlappingsubproblems,meaningthesamesubproblemsareusedtosolvemanydifferentprobleminstances,aquicker approach called dynamicprogrammingavoids recomputing solutions that have alreadybeencomputed
    - Thegreedymethod
      - ✓ Agreedyalgorithmissimilartoadynamicprogramming algorithm, but the difference isthatsolutionstothesubproblemsdonothavetobe knownateachstage,insteada“greedy”

choice can be made of what looks best for themoment.

- Linearprogramming
  - ✓ Whensolvingaproblemusinglinearprogramming ,specificinequalitiesinvolvingtheinputsarefound andthenanattemptismadeto maximizesomelinear functionoftheinputs.
- Reduction
  - ✓ Thistechniqueinvolvessolvingadifficultproblem by transforming it into a better knownproblemforwhichasymptoticallyoptimala lgorithms.
- Searchandenumeration
  - ✓ Many problems can be modeled as problems ongraphs.
  - ✓ A graph exploration algorithm specifies rules formoving around a graph and is useful for suchproblems.
  - ✓ This category also includes search algorithms,branchandboundenumerationandbac ktracking.

- **Byfieldof study**
  - **i.** Inthefieldofcomputersciencehasitsownproblemandrequiresef ficientalgorithm
  - **ii.** Relatedproblemsinone fieldareoftenstudiedtogether.
  - **iii.** Some example classes are search algorithms, sortingalgorithms,mergealgorithmsetc…..

**Sarvodaya College**

**(AFFILIATEDTOSAURASHTRAUNIVERSITY)**

| SR.NO. | QUESTION | ANSWER |
|---|---|---|
| 1 | WHICHOFTHEFOLLOWINGCASEDOESNOTEXISTINCOMPLEXITYTHEORY? | NULLCASE |
| 2 | ANALGORITHMIS_____ | APROCEDURE FOR SOLVING APROBLEM |
| 3 | AN ALGORITHM IN WHICH WE DIVIDE THEPROBLEM INTO SUBPROBLEM AND THEN WECOMBINETHESUBSOLUTIONSTOFORMSOLUTIONTOTHEORIGINALPROBLEMIS KNOWNAS_____ | DIVIDEAND CONQUER |
| 4 | ANALGORITHMWHICHUSESTHEPASTRESULTSANDUSESTHEMTOFINDTHENEWRESULTS IS _____ | DYNAMICPROGRAMMINGALGORITHMS |
| 5 | ACOMPLEXITYOFALGORITHMDEPENDS UPON_____ | TIMEAND SPACE |
| 6 | ANALGORITHMWHICHTRIESALLTHEPOSSIBILITIESUNLESSRESULTSARESATISFACTORYISANDGENERALLYISTIME-CONSUMINGIS _____ | BRUTEFORCE |
| **7** | FORARECURSIVEALGORITHM_____ | ABASECASEISNOT NECESSARY |
| **8** | FOR AN ALGORITHM WHICH IS THE MOSTIMPORTANTCHARACTERISTICTHATMAKESITACCEPTABLE_____ | CORRECTNESS ANDPRECISION |
| **9** | IFFORANALGORITHMTIMECOMPLEXITY ISGIVENBYO(1)THENTHECOMPLEXITYOFITIS _____ | CONSTANT |
| **10** | IFFORANALGORITHMTIMECOMPLEXITY | LINEAR |

| | | |
|---|---|---|
| | ISGIVENBYO(N)THENTHECOMPLEXITYOFITIS _____ | |
| **11** | WHICHALGORITHMISBETTERFORSORTINGBETWEENBUBBLESORTANDQUICKSORT? | QUICKSORT |
| **12** | PERFORMANCE BASED CRITERIA OF ALGORITHM,WHICHHASTODOWITHITSCOMPUTINGTIMEIS_____ | TIMECOMPLEXITY |
| **13** | PERFORMANCEBASEDCRITERIAOFALGORITHM, WHICH HAS TO DO WITH ITSSTORAGEIS_____ | SPACECOMPLEXITY |

## 5. Big-OhNotation

- "Big-O"notationwasintroducedinP.Bachmann's1892bookAnalytischeZahlentheorie.
- The Big O notation defines an upper bound of an algorithm, it bounds a function only from above. The Big O notation is useful when we only have upper bound on time complexity of an algorithm.
- Many times, we easily find an upper bound by simply looking at the algorithm.
  O(g(n)) = {f(n): there exist positive constants C and No such that 0 <= f(n) <= C*G(n) for all N >= No}

## 6. Big-Omega Ω Notation

Just as Big O notation provides an asymptotic upper bound on a function, Ω notation provides an asymptotic lower bound. Ω Notation can be useful when we have lower bound on time complexity of an algorithm.