

1.1 - Big O Notation

Complexity	General form	Examples
Constant	$O(n^0)$ or $O(1)$	Is number odd / even
Logarithmic	$O(\log n)$	Binary search, half each iteration
Linear	$O(n)$	Linear search
Polynomial	$O(n^k)$	$O(n^2)$, $O(n^3)$ etc. Bubble sort
Exponential	$O(2^n)$	Recursive Fibonacci algorithm
Factorial	$O(n!)$	Brute force travelling salesman problem

A theoretical measure of the execution of an algorithm, usually the time or memory needed, given the problem size n , which is usually the number of items. ... Formal Definition: $f(n) = O(g(n))$ means there are positive constants c and k , such that $0 \leq f(n) \leq cg(n)$ for all $n \geq k$.

2.1.2 - Sorting Algorithms

```

/* use insertion sort to sort x[0], x[1], ..., x[n-1] into ascending order */
for i = 1 to n-1 do
    /* insert x[i] into its proper position among x[0], x[1], ..., x[i-1] */
    temp = x[i];
    j = i-1;
    while (j >= 0 and (temp < x[j])) do
        x[j+1] = x[j];
        j = j - 1;
    end while
    x[j+1] = temp;
end for
    
```

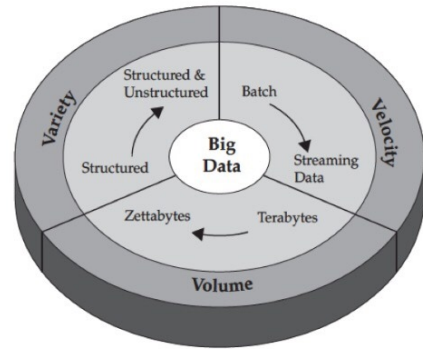
In computer science a sorting algorithm is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order.

3.1.3 - Recursion

$$a_n = a_{n-1}r$$

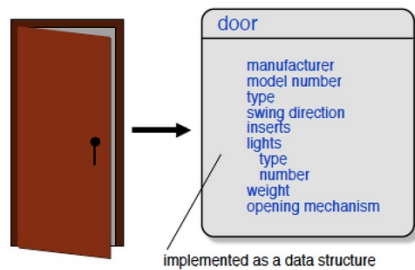
Recursion definition, the process of defining a function or calculating a number by the repeated application of an algorithm. The most common application of recursion is in mathematics and computer science, where a function being defined is applied within its own definition.

4.1.4 - Big Data



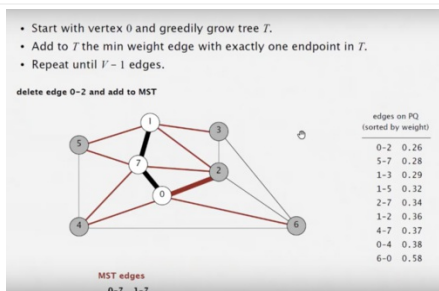
Extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions. Big data describes data sets so large and complex that is impossible to manage with conventional data processing tools.

5.1.5 - Data Structures



Array
Tree
Stack
Queue
Graph
Hash Table
Linked List
Heap

6.2.1 - Greedy Algorithm



A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. A greedy algorithm picks the best immediate choice and never reconsiders its choices.

7. 2.2 - Hill Climbing

```
function HILL-CLIMBING(problem) returns a state
  current ← make-node(problem.initial-state)
  loop do
    neighbor ← a highest-valued successor of current
    if neighbor.value ≤ current.value then
      return current.state
    current ← neighbor
```

In numerical analysis, hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution.

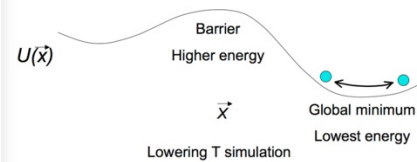
The hill climbing algorithm attempts to find a better solution by generating a neighboring solution. Each neighboring solution is generated based on the best solution so far, with a single element modified.

8. 2.3 - Simulated Annealing

Simulated Annealing

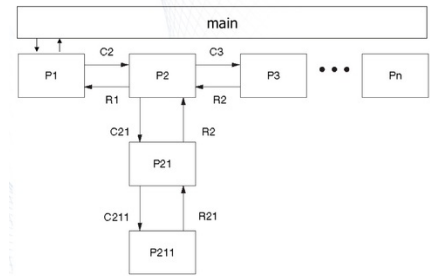
- Slowly reducing the temperature during the simulation, leads to smaller amplitude motions, less likely to jump barriers, but with more reasonable geometry

This allows the molecule to return to a more "realistic" configuration.



Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities).

9. 2.4 - Dynamic Programming



In computer science, mathematics, management science, economics and bioinformatics, dynamic programming (also known as dynamic optimization) is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions.

$1+1+1+1+1+1+1+1 = 8 \dots$ "this" plus one = 9

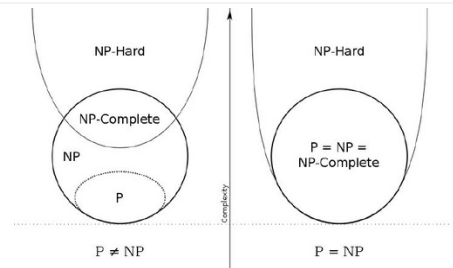
The example above describes memoization (yes memoization not memorization), a top-down approach in dynamic programming, which store the results of previous computations for future use.

10. 2.5 - Machine Learning



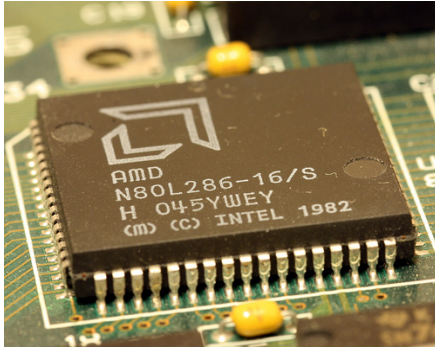
Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. ... The process of machine learning is similar to that of data mining.

11. 2.6 - P vs NP Problem



The set of NP-complete problems is often denoted by NP-C or NPC. The abbreviation NP refers to "nondeterministic polynomial time". ... As a consequence, determining whether it is possible to solve these problems quickly, called the P versus NP problem, is one of the principal unsolved problems in computer science today.

12. **3.1 - How do computers work?**



The computer does its primary work in a part of the machine we cannot see, a control center that converts data input to information output. This control center, called the central processing unit (CPU), is a highly complex, extensive set of electronic circuitry that executes stored program instructions.

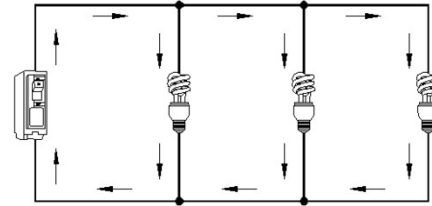
13. **3.2 - Halting Problem**



In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running or continue to run forever.

We say that a program "solves the halting problem" if it can look at any other program and tell if that other program will run forever or not.

14. **4.1 - Parallelism**



A parallel computer is a set of processors that are able to work cooperatively to solve a computational problem. This definition is broad enough to include parallel supercomputers that have hundreds or thousands of processors, networks of workstations, multiple-processor workstations, and embedded systems.

A computer program is, in essence, a stream of instructions executed by a processor. Without instruction-level parallelism, a processor can only issue less than one instruction per clock cycle ($IPC < 1$). These processors are known as subscalar processors.

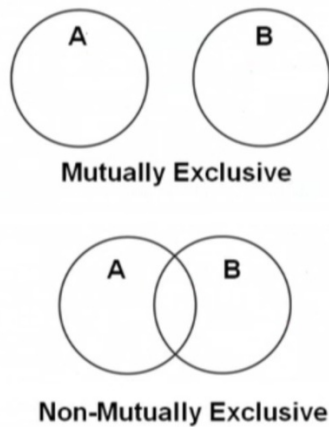
15. **4.2 - Race Condition**



A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.

A race condition (or race hazard is the behavior of an electronic, software, or other system) where the output is dependent on the sequence or timing of other uncontrollable events becomes a bug when events do not happen in the order the programmer intended.

16. **4.3 -
Mutual
Exclusion
(Mutex)**



In computer science, mutual exclusion is a property of concurrency control, which is instituted for the purpose of preventing race conditions; it is the requirement that one thread of execution never enter its critical section at the same time that another concurrent thread of execution enters its own critical section.

17. **4.4.1 -
Binary
Semaphore**



Semaphores can be binary (0 or 1) or can have additional values. Typically, a process using semaphores checks the value and then, if it using the resource, changes the value to reflect this so that subsequent semaphore users will know to wait.

18. **4.4.2 -
Counting
Semaphore**



Conceptually, a semaphore is a nonnegative integer count. Semaphores are typically used to coordinate access to resources, with the semaphore count initialized to the number of free resources. Threads then atomically increment the count when resources are added and atomically decrement the count when resources are removed.

When the semaphore count becomes zero, no more resources are present. Threads that try to decrement the semaphore when the count is zero block until the count becomes greater than zero.

19. **4.4 -
Semaphore**



In computer science, a semaphore is a variable or abstract data type used to control access to a common resource by multiple processes in a concurrent system such as a multiprogramming operating system.

A semaphore is a value in a designated place in operating system, especially Unix, (or kernel) storage that each process can check and then change. Depending on the value that is found, the process can use the resource or will find that it is already in use and must wait for some period before trying again.

Semaphores are commonly use for two purposes: to share a common memory space and to share access to files. Semaphores are one of the techniques for interprocess communication (IPC). The C programming language provides a set of interfaces or "functions" for managing semaphores.

20. **4.5 -
Deadlock**



A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource, resulting in both programs ceasing to function. The earliest computer operating systems ran only one program at a time.

21. **5.1.1 - Brute-force Attack**



Brute-force search. ... In computer science, brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

22. **5.1.2 - Social Engineering**

Social engineering at work

- Suspicious telephone call
- Unattended persons
 - Badges
 - Processes
- Door surfing
 - Watch out for people with their hands full
- Dumpster diving
- USB drive in the parking lot

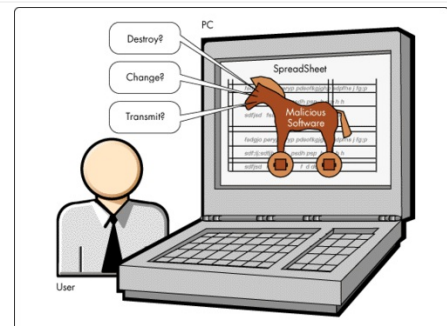
Social engineering is an attack vector that relies heavily on human interaction and often involves tricking people into breaking normal security procedures.

23. **5.1.3 - Security Exploit**



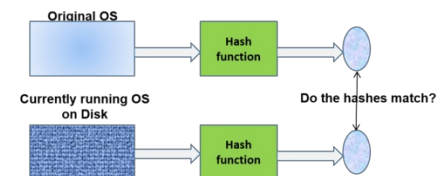
An exploit (from the English verb to exploit, meaning "using something to one's own advantage") is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability in order to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service (DoS or related DDoS)

24. **5.1.4 - Trojan Horse**



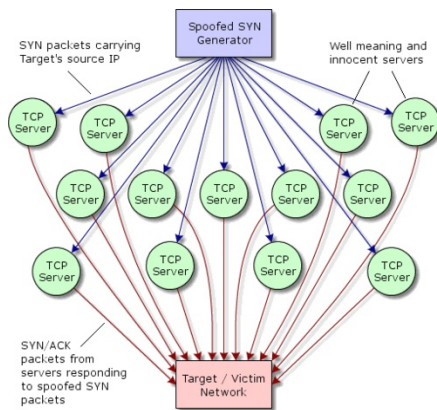
In computing, Trojan horse, or Trojan, is any malicious computer program which is used to hack into a computer by misleading users of its true intent. The term is derived from the Ancient Greek story of the wooden horse that was used to help Greek troops invade the city of Troy by stealth.

25. **5.1.5 - Rootkit**



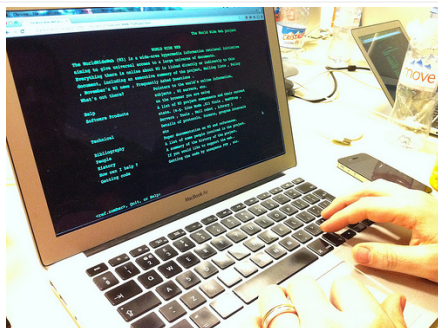
A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or areas of its software that would not otherwise be allowed (for example, to an unauthorized user) and often masks its existence or the existence of other software.

26. **5.1.6 - Distributed Denial-of-service Attack (DDoS)**



A distributed denial-of-service (DDoS) attack is an attack in which multiple compromised computer systems attack a target, such as a server, website or other network resource, and cause a denial of service for users of the targeted resource.

27. **5.1 - Computer Hacking**



Computer hacking refers to the practice of modifying or altering computer software and hardware to accomplish a goal that is considered to be outside of the creator's original objective. Those individuals who engage in computer hacking activities are typically referred to as "hackers."

28. **5.2.1 - Symmetric cryptography**



A type of encryption where the same key is used to encrypt and decrypt the message. This differs from asymmetric (or public-key) encryption, which uses one key to encrypt a message and another to decrypt the message.

29. **5.2.2 - Asymmetric cryptography**



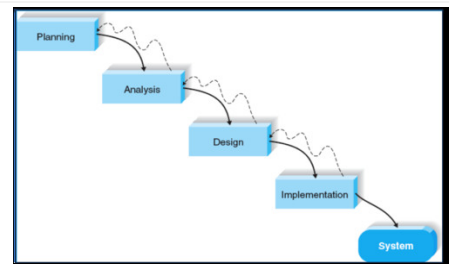
Asymmetric Encryption is a form of Encryption where keys come in pairs. ... Frequently (but not necessarily), the keys are interchangeable, in the sense that if key A encrypts a message, then B can decrypt it, and if key B encrypts a message, then key A can decrypt it.

30. **5.2 - Cryptography**



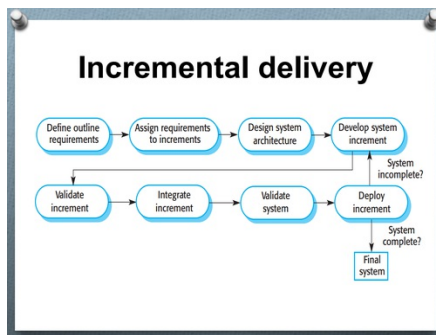
The science of coding and decoding messages so as to keep these messages secure. Coding (see encryption) takes place using a key that ideally is known only by the sender and intended recipient of the message.

31. **6.1 - Waterfall Development**



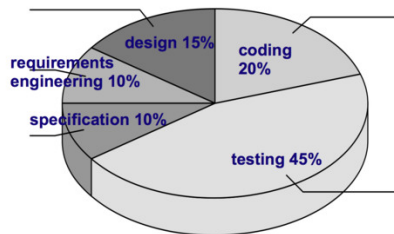
The waterfall model is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

32. **6.2 - Agile Development**



In software application development, agile software development (ASD) is a methodology for the creative process that anticipates the need for flexibility and applies a level of pragmatism into the delivery of the finished product. Agile software development focuses on keeping code simple, testing often, and delivering functional bits of the application as soon as they're ready. The goal of ASD is to build upon small client-approved parts as the project progresses, as opposed to delivering one large application at the end of the project.

33. **6.3 - Software Development In The Real World**

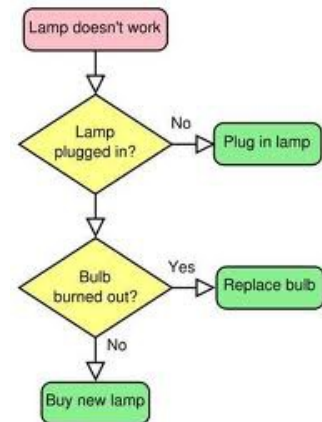


Software development is the process of computer programming, documenting, testing, and bug fixing involved in creating and maintaining applications and frameworks resulting in a software product. Software development is a process of writing and maintaining the source code, but in a broader sense, it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

34. **Common Beginner's Questions...**



35. **Core Concept #1 - Algorithms and Data Structures**



A data structure is an arrangement of data in a computer's memory or even disk storage. An example of several common data structures are arrays, linked lists, queues, stacks, binary trees, and hash tables.

An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

36. **Core Concept #2 - Artificial Intelligence**



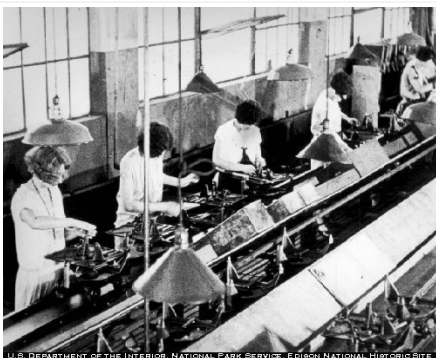
The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

37. **Core Concept #3 - Computer Architecture and Engineering**



In computer engineering, computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems. Some definitions of architecture define it as describing the capabilities and programming model of a computer but not a particular implementation.

38. **Core Concept #4 - Concurrency**



Concurrency is the ability to run several programs or several parts of a program in parallel. If a time consuming task can be performed asynchronously or in parallel, this improve the throughput and the interactivity of the program. A modern computer has several CPU's or several cores within one CPU.

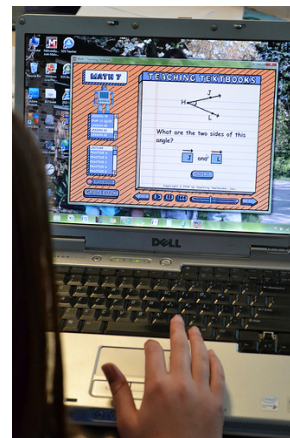
39. **Core Concept #5 - Computer Security**

Click here to enter text.

	Policy	Internal computers	Kiosk computers
Administrators			
Everyone	Allow log on locally	Authenticated Users	Guests
Guests	Allow log on through Remote Desktop Services	Administrators	Administrators
Power Users	Shut down the system	Authenticated Users	Administrators
Authenticated Users			

Computer security, also known as cyber security or IT security, is the protection of computer systems from the theft or damage to their hardware, software or information, as well as from disruption or misdirection of the services they provide.

40. **Core Concept #6 - Software Development Methodologies**



In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

41. **Descriptive Analogies for the Layman**

<http://carlcheo.com/compsci>

42. **Q1 - What is the difference between computer science and programming?**

Computer scientist is like a physicist and programmer is like an engineer.

43. **Q2 - What is programming?**

Writing very specific instructions to a dumb, yet obedient machine... You have to tell the computer what it exactly needs to do. It doesn't know how to assume and never think about the consequences.

44. **Q4 - What is the difference between Java and JavaScript?**

They are not related at all.

Java and Javascript are similar like car and carpet are similar.

45. **Q5 - What is the difference between JavaScript and JQuery?**

JQuery is a library built on top of JavaScript.

Javascript is the ugly nerd and JQuery is the wizard who turns him into the handsome quarterback.

46. Q6 - What is the difference between a framework and library?	<p>You call library. Framework calls you.</p> <p>A library is a tool. A framework is a way of life.</p>
47. Q8 - What is object-oriented programming?	<p>Objects are nouns, methods are verbs.</p> <p>Objects are like people. They're living, breathing things that have knowledge inside them about how to do things and have memory inside them so they can remember things. And rather than interacting with them at a very low-level, you interact with them at a very high level of abstraction.</p> <p>Objects encapsulate complexity, and the interfaces to that complexity are high level.</p>
48. Q9 - What is an application program interface (API)?	<p>At restaurants, you order food (call API) from the menu (APIs). Once your food is ready (API response is ready), the waiter will serve you the food.</p> <p>The basic idea is: you ask for what you want and the system returns you a response, without exposing what's happening behind the scene.</p>
49. Q10 - What is the difference between SQL and NoSQL database?	<p>NoSQL databases store information like you would recipes in a book. When you want to know how to make a cake, you go to that recipe, and all of the information about how to make that cake (ingredients, preparation, mixing, baking, finishing, etc.) are all on that one page.</p> <p>SQL is like shopping for the ingredients for the recipe. In order to get all of your ingredients into your cart, you have to go to many different aisles to get each ingredient. When you are done shopping, your grocery cart will be full of all the ingredients you had to run around and collect.</p>