Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

Topic: Running Times

Algorithm Analysis

- Space Efficiency: Somewhat Important
- Time Efficiency: **SUPER IMPORTANT!**

Running Time

`Running Time = T(n)` - N is usually the size of the input: - Number of items to sort - Number of items to search - Size of objects

- Cases:
  - Worst Case (Most Common)
  - Average Case
  - Amortized
  - Best Case

- Factors To Ignore:
  - Small Input Size
  - Speed of the Machine

Big O

Formula

$$n \geq n_0, f(n) \leq g(n)$$

Models of Computation

- A mathematical model that represents the actual computers on which algorithms will be run
- Provides a way to analyze algorithms without having to actually run them
- Examples:
  - Turning Machine (TM)
  - Random Access Machine (RAM)
  - Parallel Random Access Machine (PRAM)
- RAM: Rules for running-time analysis
  1. Each simple arithmetic operation takes constant time
  2. Each assignment takes constant time
  3. Running time of a sequence is the sum of each statement
  4. Running time of an if is the sum of all sections
  5. Running time of a loop is iterations times body
  6. Nested loops are Rule 5 from inside out

Topic: ADTs

- A description of a data structure containing:
  - I. Some information about how the data is organized (maybe)
  - II. A list of primitive operations that acess or modify the data
  - **No Implementation Details**

ADT #01- List ADT

An ordered sequence of elements (not necessarily sorted)

Primitive Operations

- `Length(list)` - Returns the number of elements in the list
- `GetFirst(list)` - Returns the first element in the list
- `GetLast(list)` - Returns the last element in the list
- `Prepend(list, x)` - Inserts x into list at the beginning
- `Append(list, x)` - Inserts x into list at the end
- `RemoveFirst(list)` - Removes the first element in the list
- `RemoveLast(list)` - Removes the last element in the list
- `CreateEmptyList()` - Reurns a newly created, empty list

- `IsEmpty(list)` - Returns `True` if list has no elements, else `False`

Implementations

Array

Description

Continuous block of memory which is not dynamically allocated. Ex. Java.

Advantages

- Easy to work with and write
- Easy access to any element within the array

Disadvantages

- O(n) time to insert new elements
- Memory allocation issues
- Can't increase size without O(n)

Linked List

Description

- There are two types of objects - Node & Header:
  - List elements are stored in the nodes
  - Header is used to access the list

Advantages

- Improved running time over array
- Solves storage problems since it's dynamic

Disadvantages

- Harder to work with and implement
- Inserting elements is still O(n)

Summary

List ADTs are used to store information in an categorized, non sorted fashion. The two implementations we studied are an Array with extra buffer space and a linked list. While the array could do most operations quickly, prepend was innefficient and the extra buffer space meant a waste of memory that didn't have a purpose. Basically memory leaking by using this method. The Linked List with a header allowed for faster times of pretty much all the operations except append which required a loop to the second to last item in the list.

Topic: Sorting Algorithms

Merge Sort

Merge Sort

- Split the array into two halves. Recursively sort each half.
- To merge 2 sorted arrays, create an empty array to hold the combined. Put a finger at the beginning of both. Copy the lesser of the two keys into the next empty cell and advance one position. Continue doing this till one array runs out. Move the rest of the array.

Quick Sort

- Partition the array so that everything in the left part is less than everything in the right part.
- Recusively sort each half. >All the work is done in the partition algorithm

Selection Sort

- Find the smallest element in the array and swap into the first cell. The first cell is now correct so don't touch it. Repeat the process until each cell is correct.

Insertion Sort

- Divided the array into sorted(left) and unsorted(right) sections. Start the size of the section sorted at 1. Take the first element following the sorted secioin and march it down into it's correct position in the sorted section, increasing the size of the sorted section by 1 for each element.

Bubble Sort

- From beginning to end, compare each element to it's neighbor, swapping if they are out of place. After each run, one more element is in its correct position at the end of the array, so repeat n *(really n-1)* times.

Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

Topic: Signifigant Figures

Rules

1. 1m = 10cm : Definitions (Infinitely Signifigant)
2. 3.94 : Nonzero Numbers (Signifigant)
3. 0.00034 : Leading Zeros (Never Signifigant)
4. 3.0094 : Captive Zeros (Signifigant)
5. Trailing Zeros:
    - 2.00 : Has Decimal (Signifigant)
    - 300 : No Decimal (Not Signifigant)

Addition / Subtraction

$$2.004 + 6.9 = 8.9$$

$$6.900 - 2.004 = 4.9$$

The answer must have the same number of digits to the right of the decimal as the number with the fewest digits to ther right of the decimal point.

Multiplication / Division

$$6.9 * 2.004 \approx 14$$

$$2.004/6.9 \approx 0.29$$

The answer must have no more sig. figs. than are in the measurement with the fewest number

Rounding (3 Sig. Figs.)

1. $6.789 \rightarrow 6.79$
    - If the last sig. fig. is followed by a >5, round up
2. $6.321 \rightarrow 6.2$
    - If the last sig. fig. is followed by a <5, round down
3. $6.55X \rightarrow 6.56$
    - If last sig. fig. is followed by a 5 with additional values, round up
4. $16.55 \rightarrow 16.6$
    - If last sig. fig. is odd and followed by only a 5, round up
5. $16.45 \rightarrow 16.4$
    - If last sig. fig. is even and followed by only a 5, round down

Topic: Elements

Week 1

- Li: Lithium
- Be: Beryllium
- H: Hydrogen
- O: Oxygen
- N: Nitrogen
- F: Flourine
- Na: Sodium
- Ne: Neon
- He: Helium
- B: Boron

Week 2

- Mg: Magnesium
- Al: Aluminum
- Si: Silicon
- P: Phosphorus
- S: Sulfer
- Cl: Chlorine
- Ar: Argon
- K: Potassium
- Ca: Calcium
- Ba: Barium

- Pu: Plutonium

Week 3

- Ti: Titanium
- Cr: Chromium
- Mn: Manganese
- Fe: Iron
- Ra: Radium
- Rn: Radon
- Co: Cobalt
- Ni: Nickel
- Cu: Copper
- Zn: Zinc
- Br: Bromine

Week 4

- Kr: Krypton
- Ag: Silver
- Cd: Cadmium
- Pt: Platinum
- Au: Gold
- Hg: Mercury
- Sn: Tin
- Pb: Lead
- I: Iodine
- Xe: Xenon
- U: Uranium

Topic: Laws

Law: Law of Conservation of Mass

- Description: Mass is neither destroyed or created during chemical reactions or physical changes

Law of Definite Proportions

- Description: A chemical compound contains the same elements in exactly the ration by mass regardless of the size, sample, or source of the compound.

Law of Multiple Proportions

- Description: If two or more different compounds are composed of the same two elements, then the ratio of the second element combined with a certain mass of the first element is always the ration of small whole #s
- Ex.
  - Carbon & Oxygen: CO or CO_2

Protons

- The number of protons in an atom's nucleus determines the atom's identity
- (Atomic Number, Z):
  - Atomic Number: # of protons & indirectly electrons
  - Z: Atomic mass
- Nuclear Force: Short-range proton-proton, neutron-neutron, and proton-neutron forces that hold the nuclear particle together. Can be referred as "Strong force".

Isotopes

- Description: Atom of the same element that differ in mass.
- They have the same number of protons but different number of neutrons
- Atomic Number: Number of protons
- Number of protons determines identity
- Number of neutrons determines mass $\rightarrow$ mass = protons + neutrons
- Revision to Dalton's Theory:
  - Although isotopes differ in mass they do not significantly differ in their chemical properties

Mass

- Description: Mass is the total amount of protons and neutrons in the nucleus of the isotope
- Ex.
  - 2 protons + 2 neutrons = 4 $\rightarrow$ Mass Number of Helium - 4

Misc.

- Nuclide: Particular kind of atom containing a definite number of protons and neutrons
- Mass #, A: Total # of nucleons(particles that make up the nucleus $\rightarrow$ protons and neutrons)

Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

Topic: Week 2

Sort

Selection Sort

- Description:
  - Find the smallest unsorted element, and add it to the end of the sorted list
  - Worst Case: $O(n^2)$
  - Best Case: $\Omega(n^2)$

Bubble Sort

- Description:
  - Move higher valued elements to the right, lower to the left
  - Keep swapping adjacent elements until the counter = 0
  - Check to see if sorted if you don't need to swap
  - Worse Case: $O(n^2)$
  - Best Case: $\Omega(n)$

Insertion Sort

- Description:
  - Go through the array and put elements in their correct place
  - For each element find within the sorted section where it belongs
  - Worse Case: $O(n^2)$
  - Best Case: $\Omega(n)$

Merge Sort

- Description:
  - Recursively sorting the array, consider each element individual
  - Sort the left half then right half
  - Worst Case: $O(n \log n)$
  - Best Case: $\Omega(n)$

Search

Linear Search

- Description:
  - Look from left to right for a specified element
  - Still works if element isn't in the array
  - Doesn't have to be a sorted array
  - Worst Case: $O(n)$
  - Best Case: $\Omega(1)$

Binary Search

- Description:
  - Array **HAS TO BE SORTED**
  - Divide and conquer, throw out half the array at a time
  - Worse Case: $O(n \log n)$
  - Best Case: $\Omega(1)$

Recursion

- Description:
  - A function that calls itself during execution
  - Base case for when to stop

Topic: Week 8

SQL

- Description: Database for storing information
- SQL stands for Structured Query Language
- Sole purpose is to query a database

Tables

- Description: Inside your database, you need to store tables.
- You'll have to make every column
- All queries query for information in each row
- Every column can store different types

Data Types

- Int: Stores integers
- SmallInt, TinyInt, MediumInt, LargeInt: Different upper bounds
- Decimal: A.K.A Doubles
- Floats: Floating point numbers
- Date: Current date
- Time: Current time
- Datetime: Specific date and time
- Timestamp: THIS SPECIFIC DATE AND TIME
- Text: A.K.A Strings
- Enums: Only can store limited set of values
- Char: Fixed length string
- VarChar: Variable-length string, also needs max length

Primary Key

- Description: Primary keys allow rows to be uniquely identified
- Also possible to have two rows that together will always be unique
- **MUST HAVE**
- Good idea to have it as an integer and have it autoincrement

Primary Operations

1. Insert:
    - Adds information to a table
    - Skeleton: `INSERT INTO <table> (<columns>) VALUES (<VALUES>)`
2. Select:
    - Extract information from a table
    - Skeleton: `SELECT <columns> FROM <table> (WHERE <condition> ORDER BY <column>)`
3. Update:
    - Modify information in a table
    - Skeleton: `UPDATE <table> SET <column> = <value> WHERE <condition/>`
4. Delete
    - Remove information in a table
    - Skeleton: `DELETE FROM <table> WHERE <condition>`

Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

Unit 1

Domain & Range

- Definition: A function from a set D to a set R is a rule that assigns to every element in D a unique in element in R.
- Familiar Definition: A function exists when every input(x) has only on output(y)
- Notation: $y = f(x)$
- Here x is the **independent variable** and y is the **dependent variable**
- To be a function the graph must pass the vertical line test which states that a graph (set of points (x,y)) in the *xy*-plane defines y as a function of x if no vertical line intersects the graph in more than one point

Domain

Given a function $y = f(x)$, the domain of the function is set of all permissible inputs and the range is the set of all resulting outputs. Domains can be found algebraically; rangges are often found algebraically and graphically. Domains and Ranges are sets. Therefore, you must use the proper notation.

Notation

- { } - Set (of intervals)
- (,) - Interval does not include the endpoint

- [,] - Intervals includes the endpoints
- $\cup$ - The union of intervals

Finding Domain

- The key to finding a domain is to find everything that can't be used.

1. Domain of all polynomial functions is the $\mathbb{R}$
2. Square root functions cannot contain a negative underneath the radical
3. Rational functions cannot have zeroes in the denominator

Continuity

- Definition: A function is continuos at a point if, graphically speaking, the graph does not come apart at that point. In other words, the graph has no breaks or holes. This graph is continuos everywhere. Notice that the graph has no breaks or holes. This means that if we are studying the behavior of the function *f* for *x* values close to any particular real number *a*, we can be assured that the *f(x)* will be close to the *f(a)*.

Types of Discontinuity

1. Removable Continuity
    - Can be patched up by re-defining the hole
2. Jump Discontinuity
    - Jump in function values that make it so you can't redefine the hole
3. Infinite Discontinuity
    - Vertical asymptote, limits so that its seperated by an asymptote

Piecewise Functions

- Definition: Piecewise functions are functions that are defined by using different equationss for different parts of the domain. Studied for discontinuity.

Increasing and Decreasing

- Definition: You can visually determine the parts of a function (intervals) where it is either increasing, decreasing, or constant.

1. Increasing: If for any two points in the interval, a positive change in *x* results in a positive change in *y*
2. Decreasing: If for any two points in the interval, a positive change in *x* results in a negative change for *y*
3. Constant: If for any two points in the interval, a positve change in *x* results in a zero change in *y*

Local and Absolute Extrema

- Definition: Extrema are the peaks and valleys where a graph changes from increasing to decreasing or vice versa. Extrema come in two forms: maxima and minima.
    - Local / Relative: Maximum or minimum on some open interval
    - Absolute: Maximum or minimum for the function

Graphs

**NOTE: THESE DON'T RENDER RIGHT ON WEBSITE ### $f(x) = x$ - Domain: $(-\infty, \infty)$ - Range: $(-\infty, \infty)$ - Increasing: $(-\infty, \infty)$ - Decreasing: N/A - Constant: N/A - Extrema: None - Symmetry: Odd - Asymptotes: None

$f(x) = x^2$

- Domain: $(-\infty, \infty)$
- Range: $[0, \infty)$
- Increasing: $[0, \infty)$
- Decreasing: $(-\infty, 0]$
- Constant: N/A
- Exrema:
  - Abs. Minima: 0 when x = 0
- Symmetry: Even
- Asymptotes: None

$f(x) = x^3$

- Domain: $(-\infty, \infty)$
- Range: $(-\infty, \infty)$
- Increasing: $(-\infty, \infty)$
- Decreasing: N/A
- Constant: N/A
- Extrema: None
- Symmetry: Odd
- Asymptotes: None

$f(x) = \frac{1}{x}$

- Domain: $(-\infty, 0)U(0, \infty)$
- Range: $(-\infty, 0)U(0, \infty)$
- Increasing: N/A
- Decreasing: $(-\infty, 0)U(0, \infty)$
- Constant: N/A
- Extrema: None
- Symmetry: Odd
- Asympototes: $x = 0, y = 0$

Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

PreUnit

Preterito

Reasons For Use

1. Defined Time Frame
2. These Cases:
    - Vivid action
    - Completed event
    - Beginning or end of an action
    - Sequence of events
3. Trigger Words:
    - Anoche
    - Ayer
    - Anteayer
    - El __ pasado
    - De repente
    - De pronto
    - Una vez

Imperfect

Reasons For Use

1. Undefined Time Frame
2. These Cases:
    - Description
    - Age or time
    - Weather
    - Habits, Used To...

    - Was/Were –ing
3. Trigger Words:
    - Mientras
    - Generalmental
    - De vez en cuando
    - A veces
    - Frecuentemente

Contents

Notes: By Nota

Nota

Nota, is a simple script to manage notes. It manages all my class notes which are markdown files and compiles them to PDF versions. This is so people can see my notes and I can share them easier. Not only that but the PDF version makes for easier reading, while the markdown versions are easy to edit and easy to search for items. Hope you enjoy!

Unit 1: Food First

Vocab

1. Prehistory: The period of time before writing was invented
2. Artifact: Objects made by humans
3. Anthropology: The study of the prehistory time period
4. Archaeology: Study of past people through material remains
5. Culture: The way of life of a society
6. Technology: Skills and tools people used to meet their basic needs and wants
7. Paleolithic Period: Period of time from 2 milllion B.C. to 10,000 B.C.
8. Neolithic Period/Revolution: Period of time from 10,000 B.C. to end of prehistory
9. Nomads: People who moved from place to place to find food
10. Hunters and Gatherers: Depended on environment for food
11. Domesticate: Raise plants/animals in controlled way for human use
12. Surplus: More than was necessary
13. Traditional Economy: An economy that relies on habit, custom, or ritual
14. Civilization: A complex, highly organized social order
15. Polythesis: Believe in many gods
16. Artisans: Skilled craftspeople
17. Cultural Diffusion: The spread of ideas, customs, and techonologies from one people to another.
18. Urbanization: The process of making an area more urban
19. Stratification: System or formation of layers, classes, or categories
20. Job Specialization: Jobs becoming increasingly specialized
21. Agriculture: The science or practice of farming
22. Agrarian Society: Any society whose economy is based on producing and maintaining crops
23. Subsitence: The act of maintaing oneself at a minimum level
24. BCE: (Before Common Era) Refers to previous to 1 CE
25. CE: (Common Era) All years after 1 CE

Agricultural Turning Point

- Life Before Farming:

1. Nomads: Move from place to place for food
2. Lived in small bands, ~20 people
3. Hunters & Gatherers
4. Made tools out of stone, bone, or wood
5. Developed spoken language
6. Fires for cooking and animal skins for clothes
7. Learned to travel across water
8. Belief in life after death
9. Animism $\rightarrow$ World is full of spirits and forces that reside in animals, objects, or dreams
10. Cave paintings were a part of religious rituals

- Life After Farming
    1. Stable food supply
    2. Longer lives
    3. Land wars
    4. New technologies like arrows
    5. Permanenet settlement
    6. Domestication
    7. New jobs
    8. Better QOL (Quality of Life)
    9. More difference in class
    10. Larger populations
    11. Large scale war