

## Contents

<b>Notes: By Nota</b>	<b>1</b>
Nota . . . . .	1
<b>Topic: Running Times</b>	<b>1</b>
Algorithm Analysis . . . . .	1
Running Time . . . . .	1
Big O . . . . .	2
Formula . . . . .	2
Models of Computation . . . . .	2
<b>Topic: ADTs</b>	<b>2</b>
ADT #01- List ADT . . . . .	2
Primitive Operations . . . . .	2
Implementations . . . . .	3

## 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:
  - Turing 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 access 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()` - Returns 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  $O(n)$

## Contents

<b>Notes: By Nota</b>	<b>1</b>
Nota . . . . .	1
<b>Topic: Significant Figures</b>	<b>1</b>
Rules . . . . .	1
Addition / Subtraction . . . . .	1
Multiplication / Division . . . . .	1
Rounding (3 Sig. Figs.) . . . . .	2

## 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: Significant Figures

### Rules

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

### 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 the 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

## Contents

<b>Notes: By Nota</b>	<b>1</b>
Nota . . . . .	1
<b>Topic: Week 2</b>	<b>1</b>
Sort . . . . .	1
Selection Sort . . . . .	1
Bubble Sort . . . . .	1
Insertion Sort . . . . .	1
Merge Sort . . . . .	2
Search . . . . .	2
Linear Search . . . . .	2
Binary Search . . . . .	2
Recursion . . . . .	2

## 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
  - Worst Case:  $O(n \log n)$
  - Best Case:  $\Omega(1)$

## Recursion

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

## Contents

<b>Notes: By Nota</b>	<b>1</b>
Nota . . . . .	1
<b>Unit 1: Food First</b>	<b>1</b>
Vocab . . . . .	1

## 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 million 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. Polytheism: Believe in many gods
16. Artisans: Skilled craftspeople
17. Cultural Diffusion: The spread of ideas, customs, and technologies 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. Subsistence: The act of maintaining 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