

DarkCybernetics Algorithms Complete Series Episode 1: Bubble Sort

By Dion Jackson

www.DarkCybernetics.com

All Rights Reserved 2017.

Video Table of Contents

Part 1. Bubble Sort and Brute Force Specifications

Part 2. Algorithm Design Paradigms

Part 3. Data Structures and Programming Techniques

Duration: (30 minutes)

Bubble Sort Algorithm

General Description: The goal of this algorithm is to arrange a fixed array or list of size N integers. The arranging process is accomplished by making an equivalence comparison between J and $J+1$ in a linear iterative process. If $J+1$ is greater than J , the two values are swapped. The algorithm terminates when the list or array values are in ascending order.

Problem Description, Define the **Bubble Sort** algorithm exactly: Use a blank sheet of paper to practice writing notation.

Coding Challenge

Implement the **Bubble Sort** algorithm in any programming language. Name , Date, and save the document with today's date and name it **FirstName_LastName_Date_EP1**.

Terms and Concepts

Define the items inside the **Terms and Concepts** section.

Fill this In!

Use scratch paper to complete the following **Problem Description** for the **Bubble Sort** algorithm provided below:

Figure 1.2 Bubble Sort Problem Description.

Name of the Algorithm:

Input:

Output:

Goal:

Algorithm's Pseudo Code:

Proof of Running time:

Proof of Correctness:

Terms and Concepts

1. Order of Growth-
2. Enigma Machine –
3. List-
4. Stack-
5. FIFO-
6. LIFO-
7. Delete-
8. Data Structure –
9. Algorithm-
10. Array-
11. Record-
12. Field-
13. File-
14. Database-
15. Variable –
16. Value –
17. Brute Force-
18. Bubble Sort –
19. Running Time-
20. Space Efficiency-
21. Upper Bound-
22. Lower Bound-
23. Control Structures-

- 24. If Loop-**
- 25. While Loop-**
- 25. IF-Else Loop-**
- 26. “Hello World”-**
- 27. Java**
- 28. C++**
- 29. Source Code –**
- 30. Terminating Statement –**
- 31. Method-**
- 32. Function-**
- 33. Inheritance-**
- 34. Polymorphism-**
- 35. OOP -**
- 36. Program-**
- 37. Storage-**
- 38. Memory-**
- 39. Batch-**
- 40. Assignment Operator-**
- 41. Invoke (Call) -**
- 42. UML-**
- 43. Comments -**
- 44. Pseudo code –**
- 45. Program –**
- 46. The Steps for Mathematical Induction -**

- 47. Real Number -
- 48. Integer-
- 49. Recurrence Relation -
- 50. Computational Complexity -
- 51. Memory-
- 52. Asymptotic Analysis-
- 53. Big O-
- 54. Big Omega-
- 55. Big Theta-
- 56. Proof Steps-
- 57. Conditional Operators -
- 58. Relational Operators-
- 59. Proof of Correctness-
- 60. Algorithm Design Process -
- 61. Brute Force -