**Computer Science
Flowcharts, Pseudocode, and Algorithms Review Sheet**

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Match the following flowchart symbol to its name**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ A) Input or Output
2. \_\_\_\_\_\_\_\_\_\_\_\_\_ B) Direction of flow
3. \_\_\_\_\_\_\_\_\_\_\_\_\_ C) Start or Stop

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ D) Connector

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ E) Process
2. \_\_\_\_\_\_\_\_\_\_\_\_\_ F) Decision

(Fill in the blank)

1. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a diagram that represents a set of instructions.

**Pseudocode Notations (Matching) -** Match the definition to the Notation

1. \_\_\_\_\_\_\_\_\_ Indicates a user will be entering something A) For
2. \_\_\_\_\_\_\_\_\_ Indicates that an output will appear on the screen B) If – then - else
3. \_\_\_\_\_\_\_\_\_ A loop (Iteration that has a condition at the beginning C) Repeat - until
4. \_\_\_\_\_\_\_\_\_ A counting loop (iteration) D) Input
5. \_\_\_\_\_\_\_\_\_ A loop (iteration) that has a condition at the end E) Output
6. \_\_\_\_\_\_\_\_\_ A decision (selection) in which a choice is made F) While

**Fill in the blank or True / False**

1. An \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a plan, a logical step-by-step process for solving a problem. Algorithms are normally written as a **flowchart** or in **pseudocode**.
2. True or False The key to any problem-solving task is to guide your thought process.
3. True or False Exploring **one specific** way of solving a problem is the **best** way to solve it
4. When designing an algorithm there are two main areas to look at:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ - What is the final goal?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ - What hurdles need to be overcome on the way to the goal

1. True or False Before an algorithm can be designed, it is important to check that the problem is completely understood.

**Matching Part II (From Khan Academy Section)**

1. Log(n)
2. no, because it's constant, and we throw out constant values.
3. binary search
4. O(1) if it's one array. one will always take up same space.
5. asymptotic analysis
6. n
7. recursive case
8. the value of the exponent
9. rate of growth
10. big-o
11. base case
12. asymptotic notation
13. factorial
14. insertion sort
15. 1
16. recursion
17. \_\_\_\_\_\_\_\_\_What's the name for the function when we're trying to count how many different orders there are for things or how many different things we can combine things?
18. \_\_\_\_\_\_\_\_\_What does n^0 equal?
19. \_\_\_\_\_\_\_\_\_What's the Big-O time of binary search?
20. \_\_\_\_\_\_\_\_\_What does n^1 equal?
21. \_\_\_\_\_\_\_\_\_What is the term for how fast a function grows with the input size?
22. \_\_\_\_\_\_\_\_\_What is a logarithmic function trying to find?
23. \_\_\_\_\_\_\_\_\_What is the notation for the lower bound?
24. \_\_\_\_\_\_\_\_\_What's it called when you drop the constant coefficients and less significant terms from rate of growth?
25. \_\_\_\_\_\_\_\_\_Does the base of a logarithm function matter in Big-O?
26. \_\_\_\_\_\_\_\_\_Which algorithm works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to one?
27. \_\_\_\_\_\_\_\_\_What kind of sort repeatedly inserts an element in the sorted subarray to its left?
28. \_\_\_\_\_\_\_\_\_What's the Big-O space of binary search?
29. \_\_\_\_\_\_\_\_\_What is a function that calls itself, until it doesn't.
30. \_\_\_\_\_\_\_\_\_How do you measure efficiency?
31. \_\_\_\_\_\_\_\_\_What do we call the first case, where we immediately know the answer, in a recursive function?
32. \_\_\_\_\_\_\_\_\_What do we call the second case, where we have to compute the same function but on a different value?
33. \_\_\_\_\_\_\_\_\_What is the notation for the upper bound?