## Recursion (Master Theorem):

Algorithms that Use Recursions:

Finding Time Complexity for Recurrences:

Note:

* c is generally used as a \_\_\_\_\_\_\_\_\_\_\_\_\_
* c \_\_\_\_\_\_ Θ(1)
* n \_\_\_\_\_\_ cn \_\_\_\_\_\_ Θ(n)

Methods for solving recurrences:

Master Theorem:

Master Theorem 1 (Easy Version): Let \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_, and let T(n) be defined on the nonnegative integers by the recurrence: T(n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, where we interpret \_\_\_\_\_\_\_\_ to mean either \_\_\_\_\_\_\_\_\_\_ or \_\_\_\_\_\_\_\_\_\_.

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Master Theorem 2 (Easy Version): Let \_\_\_\_\_\_\_\_and \_\_\_\_\_\_\_\_, and let T(n) be defined on the nonnegative integers by the recurrence: T(n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, where \_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_, where we interpret \_\_\_\_\_\_\_ to mean either \_\_\_\_\_\_ or \_\_\_\_\_\_, then T(n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Master Theorem 3 (Extension of M2 for k < 0):

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ex:

a.)

b.)

c.)

d.)

e.)

f.)

g.)