Master Theorem

- T(N) = a T(N/b) + f(N)
  - Where f(N) = O(N^k * log^p N)
- log_b a > k : O(N^{log_b a})
- log_b a = k
  - p > -1 : O(N^k * log^{p+1} N)
  - p = -1 : O(N^k * log log N)
  - p < -1 : O(N^k * 1)
- log_b a < k
  - p > -1 : O(N^k * log^p N}
  - p = -1 : O(N^k * 1)
  - p < -1 : O(N^k * 1)
- Then what? Just sub it in ez!

Master Theorem Example 1

- T(N) = a T(N/b) + f(N)
  - Where f(N) = O(N^k * log^p N)
- log_b a > k : O(N^{log_b a})
- log_b a = k
  - p > -1 : O(N^k * log^{p+1} N)
  - p = -1 : O(N^k * log log N)
  - p < -1 : O(N^k * 1)
- log_b a < k
  - p > -1 : O(N^k * log^p N}
  - p = -1 : O(N^k * 1)
  - p < -1 : O(N^k * 1)

- T(N) = 1 * T(N/2) + c
- Where f(N) = O(N^0 log^0 N)

- log_b a = 0
- k = 0

$\log^1 N$

- p = 0

1

- O(N^0 * log^{0+1} N)
- O(log N)

Log^0(N) = log(N)^0 = log (N^0) = log(1) = log_b 1 = 0 for b = any number

Master’ theorem not work on T(-N)

- T(N) = a T(N/b) + f(N)
  - Where f(N) = O(N^k * log^p N)
- log_b a > k : O(N^{log_b a})
- log_b a = k
  - p > -1 : O(N^k * log^{p+1} N)
  - p = -1 : O(N^k * log log N)
  - p < -1 : O(N^k * 1)
- log_b a < k
  - p > -1 : O(N^k * log^p N}
  - p = -1 : O(N^k * 1)
  - p < -1 : O(N^k * 1)

- T(N) = 8 * T(N/2) + Nc
- where f(N) = O(N^1 * log^0 N)

- log_b a = 3
- k = 1