Scuola universitaria professionale
della Svizzera italiana
SUPSI

Università
della
Svizzera
italiana

Istituto Dalle Molle di studi sull'intelligenza artificiale

Algorithms and Data Structures

Addendum - Master Theorem and Recursive Functions

Matteo Salani
matteo.salani@idsia.ch

Decreasing Functions

When the size of the input is decremented by a constant.

$$T(n) = a \cdot T(n - b) + f(n)$$

- $a > 0$: number of recursive calls
- $b > 0$: decrement constant of the input size
- $f(n) = O(n^k)$: computational complexity of non-recursive part

Cases:

1. If $a = 1$: $O(n^{k+1})$ or $O(n \cdot f(n))$
2. If $a > 1$: $O(n^k \cdot a^{\frac{n}{b}})$
3. If $a < 1$: $O(n^k)$

Examples

- $T(n) = T(n-1) + 1$: $O(n)$
- $T(n) = T(n-1) + n$: $O(n^2)$
- $T(n) = T(n-1) + \log(n)$: $O(n \cdot \log n)$
- $T(n) = 2 \cdot T(n-1) + 1$: $O(2^n)$
- $T(n) = 2 \cdot T(n-3) + 1$: $O(2^{\frac{n}{3}})$
- $T(n) = 3 \cdot T(n-1) + 1$: $O(3^n)$
- $T(n) = 2 \cdot T(n-1) + n$: $O(n \cdot 2^n)$

Dividing Functions

When the size of the input is divided by a constant.

$$T(n) = a \cdot T(n/b) + f(n)$$

- $a \geq 1$: number of recursive calls
- $b > 1$: dividing constant of the input size
- $f(n) = O(n^k \log^p n)$: computational complexity of non-recursive part

Considering $\log_b a$ we have Cases and SubCases:

1. If $\log_b a > k$: $O(n^{\log_b a})$
2. If $\log_b a = k$:
   - a), if $p > -1$: $O(n^k \log^{p+1} n)$
   - b), if $p = -1$: $O(n^k \log \log n)$
   - c), if $p < -1$: $O(n^k)$
3. If $\log_b a < k$:
   - a), if $p \geq 0$: $O(n^k \log^p n)$
   - b), if $p < 0$: $O(n^k)$

Examples - Case 1

- $T(n) = 2 \cdot T(n/2) + 1$
  $a = 2, b = 2, f(n) = O(1) = O(n^0 \log^0 n) \to k = 0, p = 0$
  $\log_2 2 = 1 > 0$
  $O(n^1)$
- $T(n) = 4 \cdot T(n/2) + n$
  $\log_2 4 = 2, k = 1, p = 0$
  $\log_2 4 = 2 > 1$
  $O(n^2)$
- $T(n) = 8 \cdot T(n/2) + n^2 \log n$
  $\log_2 8 = 3, k = 2, p = 1$
  $\log_2 8 = 3 > 2$
  $O(n^3)$

Examples - Case 2

- $T(n) = 2 \cdot T(n/2) + n$
  $\log_2 2 = 1, k = 1, p = 0$
  $\log_2 2 = 1 = k, p > -1 \rightarrow$ SubCase a
  $O(n \log n)$
- $T(n) = 4 \cdot T(n/2) + n^2$
  $\log_2 4 = 2, k = 2, p = 0$
  $\log_2 4 = 2 = k, p > -1 \rightarrow$ SubCase a
  $O(n^2 \log n)$
- $T(n) = 8 \cdot T(n/2) + n^3 \log^5 n$
  $\log_2 8 = 3, k = 3, p = 1$
  $\log_2 8 = 3 = k, p > -1 \rightarrow$ SubCase a
  $O(n^2 \log^6 n)$

Examples - Case 2 and 3

Case 2:

- $T(n) = 2 \cdot T(n/2) + \frac{n}{\log n}$
  $\log_2 2 = 1, k = 1, p = -1$
  $\log_2 2 = 1 = k, p = -1 \rightarrow$ SubCase b
  $O(n \log \log n)$
- $T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log^2 n}$
  $\log_2 4 = 2, k = 2, p = -2$
  $\log_2 4 = 2 = k, p < -1 \rightarrow$ SubCase c
  $O(n^2)$

Case 3:

- $T(n) = 4 \cdot T(n/2) + \frac{n^3}{\log n}$
  $\log_2 4 = 2, k = 3, p = -1$
  $\log_2 4 = 2 < k, p < 0 \rightarrow$ SubCase b
  $O(n^3)$