Analysis Tools for Data Structures and Algorithms

Hsuan-Tien Lin

Dept. of CSIE, NTU

March 09, 2021

Motivation

Properties of Good Programs

- meet requirements, correctness: basic
- clear usage document (external), readability (internal), etc.

Resource Usage/Prediction (Performance)

- efficient use of computation resources (CPU, FPU, GPU, etc.)?
  time complexity
- efficient use of storage resources (memory, disk, etc.)?
  space complexity

need: "language" for describing the complexity

Time Complexity of Matrix Addition

MATRIX-ADD($A, B, rows, cols$)

```
C = CONSTRUCT-MATRIX(rows, cols)
for i = 1 to rows
    for j = 1 to cols
        C[i, j] = A[i, j] + B[i, j]
return C
```

- inner for: $R = P \cdot cols + Q$
- total: $(S + R) \cdot rows + T$

total time needed: $P \cdot rows \cdot cols + (Q + S) \cdot rows + T$

Rough Time Complexity of Matrix Addition

$$P \cdot rows \cdot cols + (Q + S) \cdot rows + T$$

P, Q, R, S, T hard to keep track and not matter much

MATRIX-ADD($A, B, rows, cols$)

C = CONSTRUCT-MATRIX(row, col)
for $i = 1$ to $rows$
    for $j = 1$ to $cols$
        $C[i, j] = A[i, j] + B[i, j]$
return C

- inner for: $R = P \cdot cols + Q = \text{rough}(cols)$
- total: $(S + R) \cdot rows + T = \text{rough}(\text{rough}(cols) \cdot rows)$

rough time needed: $\text{rough}(rows \cdot cols)$

Asymptotic Notation

Representing “Rough” by Asymptotic Notation

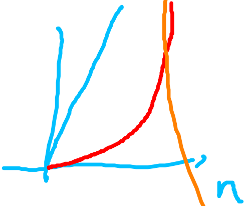
- goal: rough rather than exact steps
- why rough? constant not matter much

—when input size n large

compare two complexity functions $f(n)$ and $g(n)$

growth of functions matters

—when n large, n^3 eventually bigger than $1126n$



rough $\Leftrightarrow$ asymptotic behavior

Asymptotic Notations: Rough Upper Bound

big-O: rough upper bound

- $f(n)$ grows slower than or similar to $g(n)$: $f(n) = O(g(n))$
  小 大
  - n grows slower than n^2: $n = O(n^2)$
  - $3n$ grows similar to n: $3n = O(n)$
- asymptotic intuition (rigorous math later):

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} \leq c$$

$f(n)$ "$\leq$" $g(n)$

big-O: arguably the most used "language" for complexity