

Foundations of Computer Science

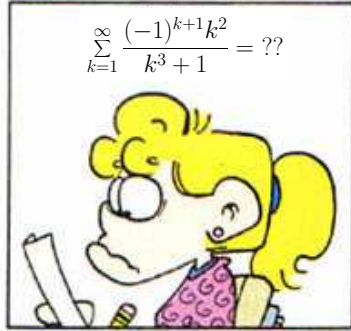
Lecture 9

Sums And Asymptotics

Computing Sums

Asymptotics: big- $\Theta(\cdot)$, big- $O(\cdot)$, big- $\Omega(\cdot)$

The Integration Method



Last Time

- 1 Structural induction: proofs about recursively defined sets.
 - ▶ Matched parentheses.
 - ▶ \mathbb{N}
 - ▶ Palindromes.
 - ▶ Arithmetic expressions.
 - ▶ Rooted Binary Trees (RBT).

Today: Sums And Asymptotics

- 1 Maximum Substring Sum
- 2 Computing Sums
- 3 Asymptotics: Big-Theta, Big-Oh and Big-Omega
- 4 Integration Method

War Story. A startup asked me about their landing webpage. Users stay 1min, or 2min (about half as many), and so on. To “convert” the 10min user is very different from “converting” the 1 min user. Who should they tailor the webpage to?

Maximum Substring Sum

1 -1 -1 2 3 4 -1 -1 2 3 -4 1 2 -1 -2 1
max. substring sum = 12

More generally, compute the maximum substring sum for

$a_1 \ a_2 \ a_3 \ a_4 \ \dots \ a_{n-1} \ a_n$ (n measures the “size” of the input).

$$T_1(n) = 2 + \sum_{i=1}^n \left[2 + \sum_{j=i}^n \left(5 + \sum_{k=i}^j 2 \right) \right]. \quad (3 \text{ for loops})$$

$$T_2(n) = 2 + \sum_{i=1}^n \left(3 + \sum_{j=i}^n 6 \right). \quad (2 \text{ for loops})$$

$$T_3(n) = \begin{cases} 3 & n = 1; \\ 2T_3(\frac{1}{2}n) + 6n + 9 & n > 1 \text{ and even;} \\ T(\frac{1}{2}(n+1)) + T(\frac{1}{2}(n-1)) + 6n + 9 & n > 1 \text{ and odd.} \end{cases} \quad (\text{recursive})$$

$$T_4(n) = 5 + \sum_{i=1}^n 10. \quad (1 \text{ for loops})$$

(What does $\sum_{i=1}^n$ mean: Pop Quiz 9.1)

Different algorithms have different runtime. Which is best?

Evaluate the Runtimes

$$T_1(n) = 2 + \sum_{i=1}^n [2 + \sum_{j=1}^n (5 + \sum_{k=1}^j 2)].$$

$$T_2(n) = 2 + \sum_{i=1}^n (3 + \sum_{j=1}^n 6)$$

$$T_3(n) = \begin{cases} 3 & n = 1; \\ 2T_3(\frac{n}{2}) + 6n + 9 & n > 1 \text{ and even;} \\ T(\frac{n}{2}(n+1)) + T(\frac{n}{2}(n-1)) + 6n + 9 & n > 1 \text{ and odd.} \end{cases}$$

$$T_4(n) = 5 + \sum_{i=1}^n 10$$

| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|----|----|----|-----|-----|-----|-----|-----|-----|-----|
| $T_1(n)$ | 11 | 29 | 58 | 100 | 157 | 231 | 324 | 438 | 575 | 737 |
| $T_2(n)$ | 11 | 26 | 47 | 74 | 107 | 146 | 191 | 242 | 299 | 362 |
| $T_3(n)$ | 3 | 27 | 57 | 87 | 123 | 159 | 195 | 231 | 273 | 315 |
| $T_4(n)$ | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 | 105 |

T_2 is better than T_1 ;

T_2 versus T_3 ???

What about T_4 ?

We need:

- Simple formulas for $T_1(n), \dots, T_4(n)$: we need to compute sums and solve recurrences.
- A way to compare runtime-*functions* that captures the essence of the algorithm.

Computing Sums: Tool 1: Constant Rule

$$S_1 = \sum_{i=1}^{10} 3 = 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 \quad 3 \times 10$$

$$S_2 = \sum_{i=1}^{10} j = j + j + j + j + j + j + j + j + j + j \quad j \times 10$$

$$S_3 = \sum_{i=1}^{10} i = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 \quad \frac{1}{2} \times 10 \times (10 + 1)$$

The *index of summation* is i in these examples.

Constants (independent of summation index) can be taken outside the sum.

$$S_1 = \sum_{i=1}^{10} 3 = 3 \sum_{i=1}^{10} 1 = 3 \times 10 \quad S_2 = \sum_{i=1}^{10} j = j \sum_{i=1}^{10} 1 = j \times 10.$$

Pop Quiz 9.2 Compute $T_4(n) = 5 + \sum_{i=1}^n 10$.

Computing Sums: Tool 2: Addition Rule

$$\begin{aligned} S &= \sum_{i=1}^5 (i + i^2) \\ &= (1 + 1^2) + (2 + 2^2) + (3 + 3^2) + (4 + 4^2) + (5 + 5^2) \\ &= (1 + 2 + 3 + 4 + 5) + (1^2 + 2^2 + 3^2 + 4^2 + 5^2) \quad (\text{rearrange terms}) \\ &= \sum_{i=1}^5 i + \sum_{i=1}^5 i^2. \end{aligned}$$

The sum of terms added together is the addition of the individual sums.

$$\sum_i (a(i) + b(i) + c(i) + \dots) = \sum_i a(i) + \sum_i b(i) + \sum_i c(i) + \dots$$

Computing Sums: Tool 3: Common Sums

$$\begin{aligned} \sum_{i=k}^n 1 &= n + 1 - k & \sum_{i=1}^n i &= \frac{1}{2}n(n+1) & \sum_{i=0}^n 2^i &= 2^{n+1} - 1 \\ \sum_{i=1}^n f(x) &= nf(x) & \sum_{i=1}^n i^2 &= \frac{1}{6}n(n+1)(2n+1) & \sum_{i=0}^n \frac{1}{2^i} &= 2 - \frac{1}{2^n} \\ \sum_{i=0}^n r^i &= \frac{1-r^{n+1}}{1-r} \quad (r \neq 1) & \sum_{i=1}^n i^3 &= \frac{1}{4}n^2(n+1)^2 & \sum_{i=1}^n \log i &= \log n! \end{aligned}$$

Example: $\sum_{i=1}^n (1 + 2i + 2^{i+2})$

$$\begin{aligned} \sum_{i=1}^n (1 + 2i + 2^{i+2}) &= \sum_{i=1}^n 1 + \sum_{i=1}^n 2i + \sum_{i=1}^n 2^{i+2} && (\text{addition rule}) \\ &= \sum_{i=1}^n 1 + 2 \sum_{i=1}^n i + 4 \sum_{i=1}^n 2^i && (\text{constant rule}) \\ &= n + 2 \times \frac{1}{2}n(n+1) + 4 \cdot (2^{n+1} - 1 - 1) && (\text{common sums}) \\ &= n + n(n+1) + 2^{n+3} - 8 && (\text{common sums}) \end{aligned}$$

Computing Sums: Tool 3: Nested Sum Rule

$$S_1 = \sum_{i=1}^3 \sum_{j=1}^3 1; \quad S_2 = \sum_{i=1}^3 \sum_{j=1}^i 1.$$

To compute a nested sum, start with the innermost sum and proceed outward.

$$S_1 = \sum_{j=1}^3 1 + \sum_{j=1}^3 1 + \sum_{j=1}^3 1 = 3 + 3 + 3 = 9.$$

$$S_2 = \sum_{j=1}^1 1 + \sum_{j=1}^2 1 + \sum_{j=1}^3 1 = 1 + 2 + 3 = 6.$$

More generally:

$$S(n) = \sum_{i=1}^n \sum_{j=1}^i 1 = \sum_{i=1}^n \sum_{\substack{j=1 \\ f(i)=i}}^i 1 = \sum_{i=1}^n i = \frac{1}{2}n(n+1).$$

Computing a Formula for $T_2(n) = 2 + \sum_{i=1}^n \left(3 + \sum_{j=i}^n 6\right)$

$$\begin{aligned} T_2(n) &= 2 + \sum_{i=1}^n \left(3 + \sum_{j=i}^n 6\right) && \text{(sum rule)} \\ &= 2 + \sum_{i=1}^n 3 + \sum_{i=1}^n \sum_{j=i}^n 6 && \text{(constant rule)} \\ &= 2 + 3 \sum_{i=1}^n 1 + \sum_{i=1}^n \sum_{j=i}^n 6 && \text{(common sum)} \\ &= 2 + 3n + \sum_{i=1}^n \sum_{j=i}^n 6 && \text{(innermost sum)} \\ &= 2 + 3n + \sum_{i=1}^n \sum_{j=i}^n 6 && \text{(constant rule)} \\ &= 2 + 3n + 6 \sum_{i=1}^n \sum_{j=i}^n 1 && \text{(constant rule)} \\ &= 2 + 3n + 6 \sum_{i=1}^n (n+1-i) && \text{(common sum)} \\ &= 2 + 3n + 6(n + (n-1) + \dots + 1) \\ &= 2 + 3n + 6 \times \frac{1}{2}n(n+1) && \text{(common sum)} \\ &= 2 + 6n + 3n^2 && \text{(algebra)} \end{aligned}$$

Practice: Compute a Formula for the Sum $\sum_{i=1}^n \sum_{j=1}^i ij$

$$\sum_{i=1}^n \sum_{j=1}^i ij = \sum_{i=1}^n \sum_{j=1}^i ij \quad \text{(innermost sum)}$$

$$= \sum_{i=1}^n i \sum_{j=1}^i j \quad \text{(constant rule)}$$

$$= \sum_{i=1}^n i \times \frac{1}{2}i(i+1) \quad \text{(common sum)}$$

$$= \frac{1}{2} \sum_{i=1}^n (i^3 + i^2) \quad \text{(algebra, constant rule)}$$

$$= \frac{1}{2} \sum_{i=1}^n i^3 + \frac{1}{2} \sum_{i=1}^n i^2 \quad \text{(sum rule)}$$

$$= \frac{1}{8}n^2(n+1)^2 + \frac{1}{12}n(n+1)(2n+1) \quad \text{(common sums)}$$

$$= \frac{1}{12}n + \frac{3}{8}n^2 + \frac{5}{12}n^3 + \frac{1}{8}n^4 \quad \text{(algebra)}$$

Summary of Maximum Substring Sum Algorithms

Runtimes

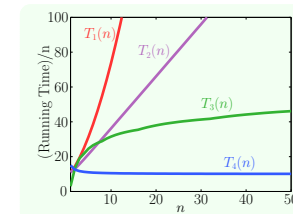
$$\mathbf{T}_1(n) = 2 + \frac{31}{6}n + \frac{7}{2}n^2 + \frac{1}{3}n^3$$

$$\mathbf{T}_2(n) = 2 + 6n + 3n^2$$

$$3n(\log_2 n + 1) - 9 \leq \mathbf{T}_3(n) \leq 12n(\log_2 n + 3) - 9$$

$$\mathbf{T}_4(n) = 5 + 10n$$

("simple" formulas for $T_1(n), \dots, T_4(n)$)



So, which algorithm is best?

Computers solve problems with big inputs. We care about large n .

- Compare runtimes *asymptotically* in the input size n . That is $n \rightarrow \infty$.
- Ignore additive and multiplicative constants (minutia). We care about *growth rate*.

Algorithm 4 is *linear* in n , $\frac{T_4(n)}{n} \rightarrow \text{constant}$.

Asymptotically Linear Functions: $\Theta(n)$, big-Theta-of- n

$$T \in \Theta(n), \text{ if there are positive constants } c, C \text{ for which } c \cdot n \leq T(n) \leq C \cdot n.$$

$$\frac{T(n)}{n} \xrightarrow{n \rightarrow \infty} \begin{cases} \infty & T \in \omega(n), & "T > n"; \\ \text{constant} > 0 & T \in \Theta(n), & "T = n"; \\ 0 & T \in o(n), & "T < n". \end{cases}$$

Linear means in $\Theta(n)$:

$$2n + 7, \quad 2n + 15\sqrt{n}, \quad 10^9 n + 3, \quad 3n + \log n, \quad 2^{\log_2 n + 4}.$$

Not linear, not in $\Theta(n)$:

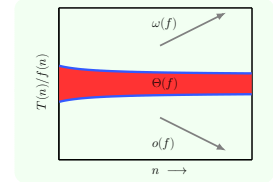
$$10^{-9} n^2, \quad 10^9 \sqrt{n} + 15, \quad n^{1.0001}, \quad n^{0.9999}, \quad n \log n, \quad \frac{n}{\log n}, \quad 2^n.$$

Other runtimes from practice:

| | | | | | | | | |
|------------------|-------------|--------------------|---------------|---------------|----------------------|---------------|--------------|---------------|
| log | linear | loglinear | quadratic | cubic | superpolynomial | exponential | factorial | BAD |
| $\Theta(\log n)$ | $\Theta(n)$ | $\Theta(n \log n)$ | $\Theta(n^2)$ | $\Theta(n^3)$ | $\Theta(n^{\log n})$ | $\Theta(2^n)$ | $\Theta(n!)$ | $\Theta(n^n)$ |

General Asymptotics: $\Theta(f)$, big-Theta-of- f

$$\frac{T(n)}{f(n)} \xrightarrow{n \rightarrow \infty} \begin{cases} \infty & T \in \omega(f), \text{ "T < f";} \\ \text{constant} > 0 & T \in \Theta(f), \text{ "T = f";} \\ 0 & T \in o(f), \text{ "T < f".} \end{cases}$$



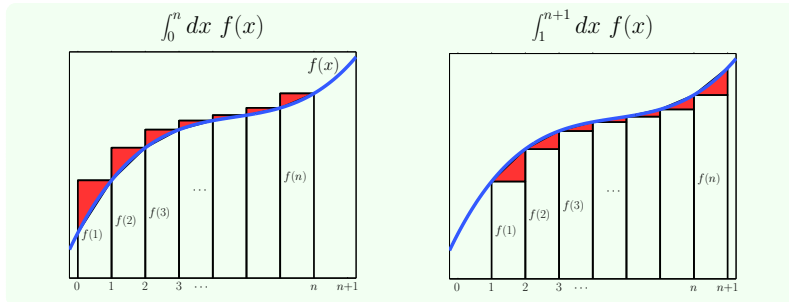
| | | | | |
|-------------------|------------------------------|-------------------|-------------------|-------------------|
| $T \in o(f)$ | $T \in O(f)$ | $T \in \Theta(f)$ | $T \in \Omega(f)$ | $T \in \omega(f)$ |
| "T < f" | "T ≤ f" | "T = f" | "T ≥ f" | "T > f" |
| $T(n) \leq Cf(n)$ | $Cf(n) \leq T(n) \leq Cf(n)$ | $Cf(n) \leq T(n)$ | | |

Examples and Practice. (See also Exercise 9.6)

- For polynomials, growth rate is the highest order.
- For nested sums, growth rate is number of nestings plus order of summand.

| | | | | | | |
|---------------|-------------------|----------------------|-------------------------------|------------------|-------------------------------|--------------------------------|
| $2n^2$ | $n^2 + n\sqrt{n}$ | $n^2 + \log^{256} n$ | $n^2 + n^{1.99} \log^{256} n$ | $\sum_{i=1}^n i$ | $\sum_{i=1}^n \sum_{j=1}^i 1$ | $\sum_{i=1}^n \sum_{j=1}^i ij$ |
| $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^4)$ |

The Integration Method



Theorem. (Integration Bound)

For a monotonically increasing function f ,

$$\int_{m-1}^n dx f(x) \leq \sum_{i=m}^n f(i) \leq \int_m^{n+1} dx f(x).$$

(If f is monotonically decreasing, the inequalities are reversed.)

Integration For Quickly Getting Asymptotic Behavior

- Integer Powers.** Set $f(x) = x^k$:

$$\sum_{i=1}^n i^k \approx \int_0^n dx x^k = \frac{n^{k+1}}{k+1} \in \Theta(n^{k+1}).$$

- Harmonic Numbers.** Set $f(x) = 1/x$ (monotonically decreasing):

$$\frac{\int_1^{n+1} dx \frac{1}{x}}{\ln(n+1)} \leq H_n = \sum_{i=1}^n \frac{1}{i} \leq \frac{1 + \int_1^n dx \frac{1}{x}}{1 + \ln n}.$$

- Stirling's Approximation for $\ln n!$.** Set $f(x) = \ln x$:

$$\ln n! = \sum_{i=1}^n \ln i \leq \int_1^{n+1} dx \ln x = (n+1) \ln(n+1) - n \in \Theta(n \ln n).$$

- Analyzing a Recurrence** $T_1 = 1; \quad T_n = T_{n-1} + n\sqrt{n} - \ln n.$

First unfold the recurrence

$$T_n = T_{n-1} + n\sqrt{n} - \ln n$$

$$T_{n-1} = T_{n-2} + (n-1)\sqrt{n-1} - \ln(n-1)$$

⋮

$$T_3 = T_2 + 3\sqrt{3} - \ln 3$$

$$T_2 = T_1 + 2\sqrt{2} - \ln 2$$

$$+ T_n = 1 + 2\sqrt{2} + \dots + n\sqrt{n} - (\ln 2 + \ln 3 + \dots + \ln n)$$

$$= \sum_{i=1}^n i\sqrt{i} - \sum_{i=1}^n \ln i$$

$$T(n) = \underbrace{\sum_{i=1}^n i\sqrt{i}}_{\Theta(n^{5/2})} - \underbrace{\sum_{i=1}^n \ln i}_{\ln n! \in \Theta(n \ln n)}$$

$$T(n) \in \Theta(n^{5/2}).$$