Run Time Analysis

- Iterative algorithm ⟶ Loop counting
- Recursive algorithm ⟶ Recurrence relations

Loop counting

Modified-Bubblesort(S)

1 + → SWAP=T

$\Sigma_{i=2..n}$ (1+ → For i=n down to 2

1+ → If SWAP=F then return

1+ → Flag=F

$\Sigma_{j=1..i-1}$ (1+ → For j=1 to i-1

3) → If $s_j > s_{j+1}$ then swap(s_j, s_{j+1}) and set SWAP=T

Return

$1+\Sigma_{i=2...n}(1+\Sigma_{j=1...i-1}\ 1))$

actually $c_1 + \Sigma_{i=2...n}(c_2+\Sigma_{j=1...i-1}c_3)$

Summation

$\Sigma_{i=2...n}\ \Sigma_{j=1...i-1} c$ $= c\Sigma_{i=2...n}(i-1)$

$= c(\Sigma_{i=1...n}(i-1)) - c$

$= c(\Sigma_{i=1...n} i - \Sigma_{i=1...n} 1) - c$

$= c(n(n+1)/2 - n) - c$

$= \mathbf{O(n^2)}$

Series

- A series is a summation of terms
- Common series
  - Arithmetic series: 1+2+...+n
  - Geometric series: $1+a+a^2+...+a^n$

Series
Things we want to do:

- Solve exactly
- Bound above or below
- Prove a solution (or bound) is correct

Closed form solutions to some common series

- $f(n) = 1+2+ ... + n = n(n+1)/2$
- $f(n) = 1+a+a^2+...+a^n = n$ if a=1
  $= (a^{n+1}-1)/(a-1)$ else

Bounds on series

For any constant k:

$$\Sigma_{i=1..n} i^k \leq \Sigma_{i=1..n} n^k$$
$$= (n^{k+1})$$
$$= O(n^{k+1})$$

Recursive Algorithms

What about Sort2?

```
Sort2(S)
    If ||S||≤1
        Return:  S
    Else
        Return:  Sort2(S\max-element(S)),max-element(S)
```

The number of steps of Sort2 satisfies:

$T(1) = c_2$
$T(n) = c_1 + T(n-1)$, $n>1$

Recurrence Relations

Methods to solve or bound:

- Guess and prove
- Unwinding
- Master method
- **WORK TREES**

Guess and Prove

- T(0) = 1
- T(1) = 1
- T(n) = T(n -1) + T(n -2)

- Closed form: TO DO

Unwinding

- T(1)=1
- T(n)= 2T(n/2) + n
  = 2(2T(n/4) + n/2) + n
  = 2(2(T(n/8) + n/4) + n/2) + n

Master Theorem

- Read the book

Recursive Algorithm A

- On input size $n \leq d$, algorithm A performs c steps.
- On input size $n > d$, algorithm A makes k recursive calls on inputs of size $n_1, n_2, \ldots, n_k$ and then takes f(n) steps to produce its solution.

Work Trees for A
n≤d

Work Trees for A
n>d