Fundamental Algorithms

Solving Recurrences

CLRS Reading: Sections 4.1, 4.2, 4.3, pages 62 -- 75
Problem Set: Assignment #2 due Friday, February 15

Fundamental Algorithms

Selection-Sort(*i*,*A*)

```
Selection-Sort(i,A)
    if i < length[A]
        then exchange A[i] ↔ A[min-Index(i,A)]
             Selection-Sort(i+1,A)
```

*Where `min-Index(i,A)` returns the index of an occurrence of the least element in *A*[*i*..*length*[*A*]].

U	N	S	O	R	T	E	D
D	N	S	O	R	T	E	U
D	E	S	O	R	T	N	U
D	E	N	O	R	T	S	U
D	E	N	O	R	T	S	U
D	E	N	O	R	T	S	U
D	E	N	O	R	S	T	U
D	E	N	O	R	S	T	U
D	E	N	O	R	S	T	U

Solving Recurrences: The Iteration Method

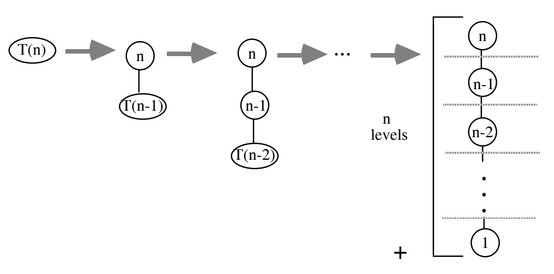
1. Iteratively construct a recursion tree by unwinding the recurrence equation.

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n-1) + n & \text{if } n > 1 \end{cases}$$

2. Determine the cost of the entire tree by summing the cost of the nodes.

D - 3

$T(n) = T(n-1) + n$

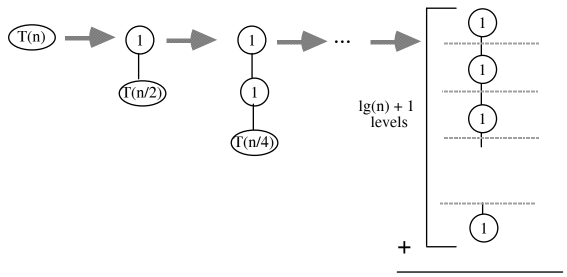


*Unless stated to the contrary, we assume the base case is a constant.

D - 4

Fundamental Algorithms

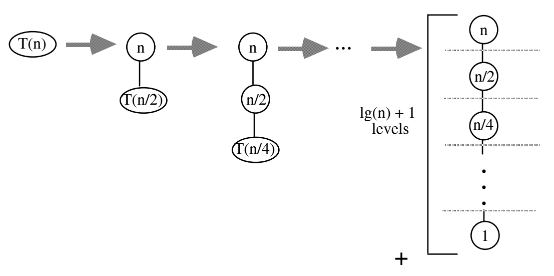
$T(n) = T(n/2) + 1$



*How many times can you divide n by 2 before you obtain a number less than or equal to 1?

Fundamental Algorithms

$T(n) = T(n/2) + n$



Special Geometric Sums and Series

$$n/2^0 + n/2^1 + n/2^2 + \ldots + n/2^{lg\,n} = \sum_{0 \le i \le lg\,n} n/2^i$$

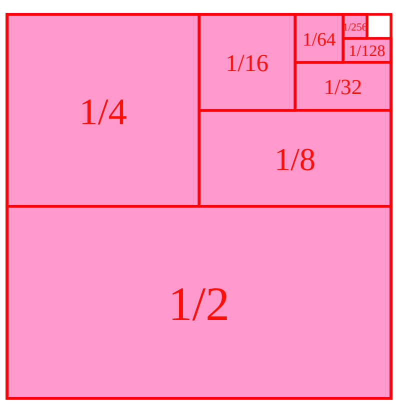
$$= n * \sum_{0 \le i \le lg\,n} (1/2)^i$$

$$\le n * \sum_{0 \le i \le \infty} (1/2)^i$$

$$=$$

Visualize It!

$$\sum_{1 \le i \le \infty} (1/2)^i$$



Geometric Sums: $\sum_{0 \le i \le k} x^i$

$$x - 1 \,\big)\overline{\; x^{k+1} \qquad\qquad\qquad\qquad - 1 \;}$$

D - 9

Geometric Sums and Series in General

- $\sum_{0 \le i \le k} x^i \;=\; \dfrac{x^{k+1} - 1}{x - 1}$
- $\sum_{0 \le i \le \infty} x^i \;=\; \lim_{k \to \infty} \sum_{0 \le i \le k} x^i$

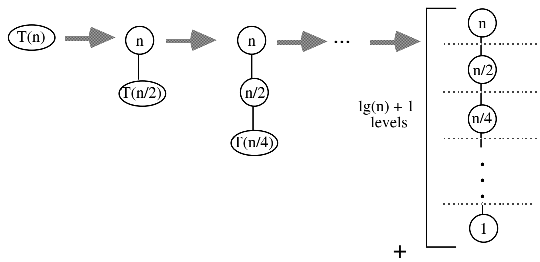
$$= \lim_{k \to \infty} \frac{x^{k+1} - 1}{x - 1} \qquad \text{if } 0 < x < 1$$

$$= \frac{-1}{x - 1}$$

$$= \frac{1}{1 - x}$$

D - 10

Getting it Exact: T(n) = T(n/2) + n



+

$\sum_{0 \le i \le lg\, n} n/2^i \;=\; n * \sum_{0 \le i \le lg\, n} (1/2)^i$

$=$

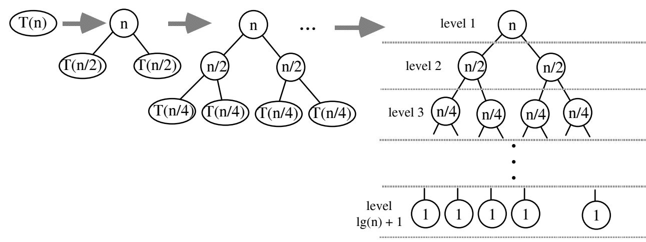
*Recall $\sum_{0 \le i \le k} x^i = \dfrac{x^{k+1} - 1}{x - 1}$

Merge-Sort Revisited

```
Merge-Sort(A, lo, hi)
    if lo < hi
        then mid ← ⌊(lo + hi)/2⌋
             Merge-Sort(A, lo, mid)
             Merge-Sort(A, mid+1, hi)
             Merge(A, lo, mid, hi)
```

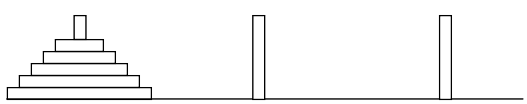
Fundamental Algorithms

$T(n) = 2*T(n/2) + n$



Fundamental Algorithms

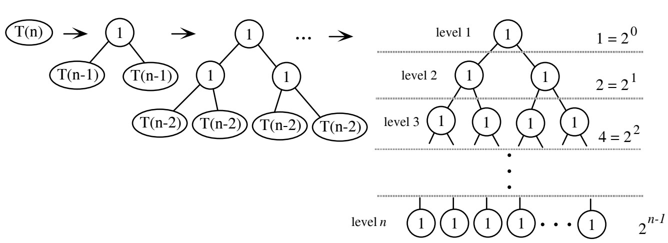
Towers of Hanoi



Shifting Sands

```
tower(n, start, finish)
    if n > 0
        then temp ← 6 - (start+finish)
             tower(n-1, start, temp)
             moveDisk(start, finish)
             tower(n-1, temp, finish)
```


$T(n) = 2*T(n-1) + 1$



*Moral: It’s better to divide than subtract.