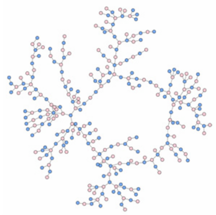


CS 173: Discrete Structures

Eric Shaffer

Office Hour: Wed. 12-1, 2215 SC

shaffer1@illinois.edu

Announcements

- Quiz Wed. 3/18
  - Covers through solving recurrences by unrolling
- Today:
  - Solving recurrence relations (Section 7.1)
    - Approximation
    - Recurrence trees
  - Algorithms (Section 3.3)
    - Analyzing theoretical efficiency
    - Searching

Solving recurrence relations

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

$T(0) = 0$

Technique: UNROLLING THE RECURRENCE

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

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

$= T(n-i) + (n-i+1)^2 + \ldots + n^2$

We reach the base case when i=n

$= T(0) + 1^2 + 2^2 + 3^2 + \ldots + n^2$ = sum of the first n squares

A few quick words about sums…

- There are formulas for some finite summations you should know

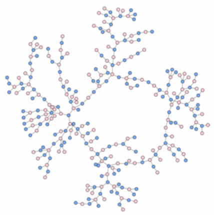
- You should know sum of first n integers
  - You can use it to solve sum of first n even numbers
  - You can use it to solve sum of first n odd numbers

- **Sum of an arithmetic series**
  - Series has constant difference **d** between terms
  - E.g. 1 + 4 + 7 + 10 + ….+ 82
  - Sum = $n(a_1+a_n)/2$

- **Sum of a geometric series**
  - Series has a constant ratio r between successive terms
  - E.g. 1 + 2 + 4 + 8 + …+ 512
  - Sum = $(a_1(r^{n+1} - 1))/(r-1)$

- Everything else you should just look up

  …like the sum of the first n squares…



Solving recurrence relations

Sum of the first n squares:

Note $(x + 1)^3 = x^3 + 3x^2 + 3x + 1$

$$1^3 = (0 + 1)^3 = 0^3 + 3(0^2) + 3(0) + 1$$
$$2^3 = (1 + 1)^3 = 1^3 + 3(1^2) + 3(1) + 1$$
$$3^3 = (2 + 1)^3 = 2^3 + 3(2^2) + 3(2) + 1$$

etc.

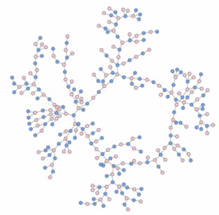
$$(n+1)^3 = (n+1)^3 = n^3 + 3n^2 + 3n + 1$$

Add up along each column

Subtract sum of the first n cubes from side

Solve for sum of first n squares

Sum of first n squares = $n(n+1)(2n+1)/6$



Divide-and-conquer recurrences

In general: $T(n) = aT(n/b) + f(n)$

Let's try a specific example: $T(n) = 2T(n/2) + n$ with $T(1) = 1$
Technique: UNROLLING THE RECURRENCE

$T(n) = 2T(n/2) + n$
$T(n) = 2(2T(n/4) + n/2) + n = 4T(n/4) + n + n$

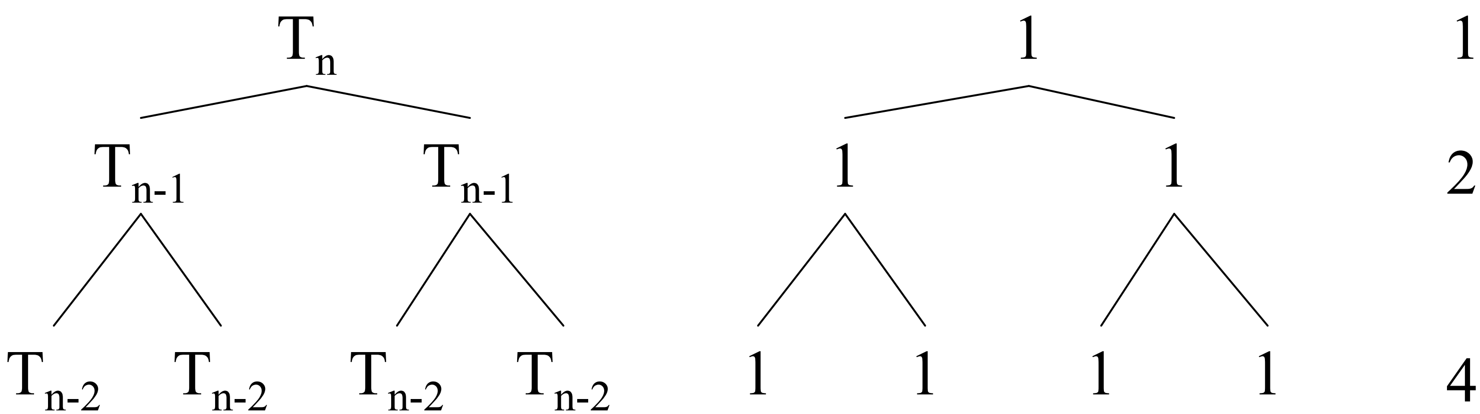
$T(n) = 2(2(2T(n/8))) + 3n$

$T(n) = 2^iT(n/2^i) + ni$

Assume n is a power of 2…when do we hit the base case? $i = \lg n$
$T(n) = n + n \lg n$

Recursion Trees

$T_n = 2\,T_{n-1} + 1$, $T_0 = 0$…let's draw a picture

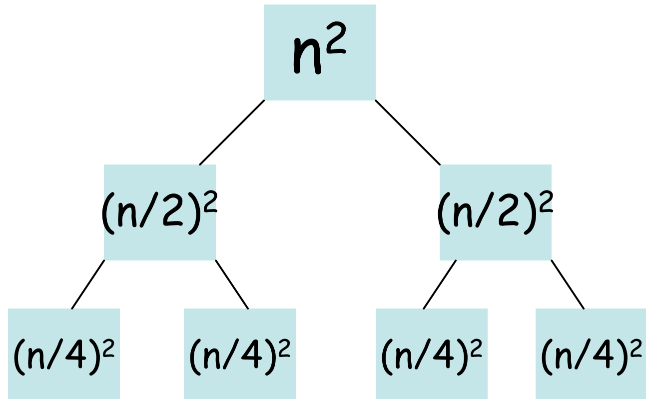


Solution is the sum of all the labels
Each label in the diagram contributes "1" to the sum

Last level of the tree will have 2^n T_0 labels
Summing by levels: $T(n) = 1+2+4+\ldots+2^{n-1} = 2^n - 1$

Approximate solutions with recursion trees

$T_n = 2T_{n/2} + n^2$, $T_1 = 1$



Sum along each level

N^2

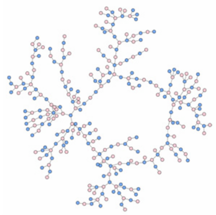
$(n^2)/2$

$(n^2)/4$

Assume n is a power of 2
There are (lg n)+1 levels in the tree

$T(n) = n^2\Sigma(1/2^i)$ with i=0,…,lgn

A decreasing geometric series dominated by first term so $T(n)=\Theta(n^2)$



Solving recurrence relations approximately

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

$T(0)=0, T(1)=1$

If we only care about approximate solutions we can observe the following

$T(n-2) \leq T(n-1)$ so $T(n) \leq 2T(n-1)$

$2T(n-1) \leq 2^n$ when we let $T(1) = T(0)=1$

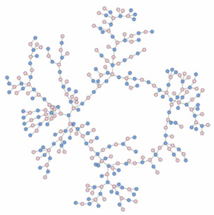
So $T(n) = O(2^n)$

Solving recurrence relations approximately

$T(n) = T(n-1) + n$
$T(0)=0, T(1)=1$

If we only care about approximate solutions we can observe the following

$T(n) = T(n-1) + O(n)$
The final sum, when we reach base case, has n terms.
So $T(n) = n\,O(n) = O(n^2)$



Solving recurrence relations approximately

So, quick summary:

1) Unrolling is best used on relations with a single recursive term

$$T(n) = T(n-a) + f(n) \quad \text{or} \quad T(n) = T(n/a) + f(n)$$

2) Trees are easier to use on relations with multiple terms

$$T(n) = c_1T(n/a_1) + c_2T(n/a_2) + f(n)$$

1) "Divide and Conquer" relations will usually have a logarithmic solution