Recurrence Relations

Def: an equation that recursively defines a sequence of values based on some initial terms.

2, 4, 6, 8, 10, 12, ... positive even integers

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

$$T(1) = 2$$

0, 1, 1, 2, 3, 5, 8, 13, ... Fibonacci

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

$$T(0) = 0 \quad T(1) = 1$$

0, 1, 3, 6, 10, 15, ...

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

$$T(0) = 0$$

1, 3, 7, 15, 31, 63, ...

$2^n - 1$

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

$$T(0) = 1$$

Backward Substitution

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

$$T(0) = 0$$

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

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

$$= T(n-3) + (n-2) + (n-1) + n$$

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

$$= T(n-n) + 1 + 2 + 3 + \dots + (n-1) + n$$

$$= 0 + 1 + 2 + 3 + \dots + (n-1) + n$$

$$= \sum_{i=0}^{n} i = \frac{n(n+1)}{2} \in \Theta(n^2)$$

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

$$T(0) = 1$$

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

$T(n-2) = 2T(n-3) + 1$

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

$$= 4T(n-2) + 2 + 1$$

$$= 4[2T(n-3)+1] + 2 + 1$$

$$= 8T(n-3) + 4 + 2 + 1$$

$$= 16T(n-4) + 8 + 4 + 2 + 1$$

$$= 2^i T(n-i) + 2^{i-1} + 2^{i-2} + \dots + 2 + 1$$

$$= 2^n T(0) + 2^{n-1} + 2^{n-2} + \dots + 2 + 1$$

$$= 2^n + 2^{n-1} + 2^{n-2} + \dots + 2 + 1$$

$$= \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T(1) = 1$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2}$$

$$T(n) = 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$$

$$= 4T\left(\frac{n}{4}\right) + n + n$$

$$= 4\left[2T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n$$

$$= 16T\left(\frac{n}{16}\right) + 4n$$

$$= 2^i T\left(\frac{n}{2^i}\right) + in$$

$$\frac{n}{2^i} = 1$$

$$2^i = n$$

$$i = \lg n$$

$$= 2^{\lg n} T\left(\frac{n}{n}\right) + n \lg n$$

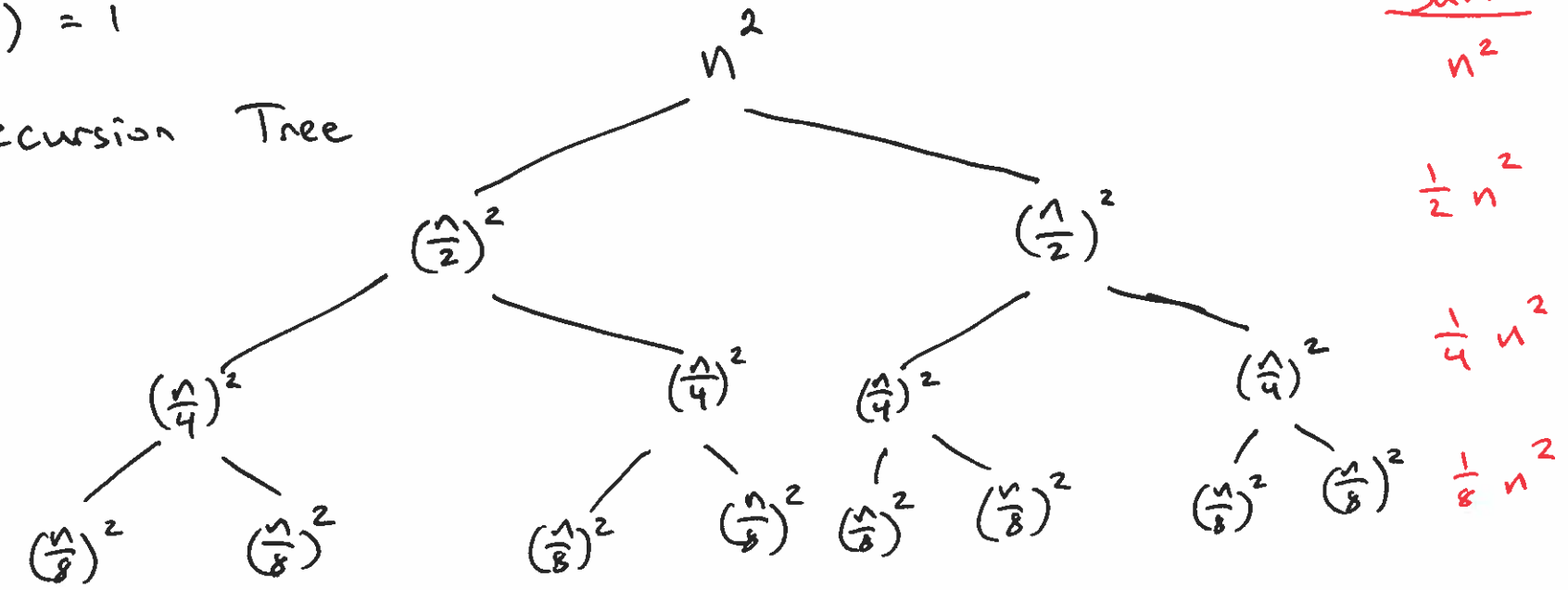
$$= 2^{\lg n} + n \lg n = n + n \lg n$$

$T(n) = 2T(\frac{n}{2}) + n^2$

$T(1) = 1$

$T(\frac{n}{2}) = 2T(\frac{n}{4}) + (\frac{n}{2})^2$

Recursion Tree



Sum

n^2

$\frac{1}{2} n^2$

$\frac{1}{4} n^2$

$\frac{1}{8} n^2$

1 ...

$$\sum_{i=0}^{L} (\frac{1}{2})^i n^2 < \sum_{i=0}^{\infty} (\frac{1}{2})^i n^2$$

$$< n^2 \sum_{i=0}^{\infty} (\frac{1}{2})^i$$

$$< 2n^2$$

Analyzing Recursive Algorithms

1. Decide on a parameter or parameters to measure the input size
2. Identify the basic operation
3. Determine if Best/Worst/average cases are different.
4. Describe a recurrence relation for the number of basic operations.
5. Solve the recurrence relation and determine the order of growth.

```
FACTORIAL(n)
1  if n = 0
2        return 1
3  return n × FACTORIAL(n − 1)
```

Input size: value of n

Basic op: multiplication

Recurrence relation:

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

$$T(0) = 0$$

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

$$T(0) = 0$$

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

$$= T(n-3) + 1 + 1 + 1$$

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

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

$$= n \in \Theta(n)$$

```
BinarySearch(A, T, low, high)
1  if high < low
2        return −1
3  mid = (low + high)/2
4  if T < A[mid]
5        return BinarySearch(A, T, low, mid − 1)
6  elseif T > A[mid]
7        return BinarySearch(A, T, mid + 1, high)
8  return mid
```

$T(n) = T\left(\frac{n}{2}\right) + a$

$T(0) = 0$

$T(1) = a$

Input size: high − low = n

Basic op: comparison between T and A[mid]

Best/Worst/Average: Best: T = A[mid] during the first iteration O(1)

Worst: T is not in A

$$T(n) = T\left(\frac{n}{2}\right) + a$$

$$T(1) = a$$

$$T(n) = T\left(\frac{n}{4}\right) + a + a$$

$$= T\left(\frac{n}{8}\right) + 3a$$

$$= T\left(\frac{n}{16}\right) + 4a$$

$$= T\left(\frac{n}{2^i}\right) + ai$$

$$= T(1) + a \lg(n)$$

$$= a + a \lg(n) \in \Theta(\log n)$$

```
POWER(x, p)
1  if p = 0
2        return 1
3  if p = 1
4        return x
5  if p is even
6        return POWER(x × x, p/2)
7  return x × POWER(x * x, p/2)
```

```
HEIGHT(T)
    // Input: A balanced binary tree T
    // Output: The height of T
1   if T is empty
2       return -1
3   return 1 + max(HEIGHT(T.left), HEIGHT(T.right))
```

What is the asymptotic complexity of this algorithm?