Introduction to Recurrence Relations

A useful mathematical fact

$$x^{\log_b y} = y^{\log_b x}$$

Can you see why this is true for all positive values of x, y, and b?

Max contiguous subsequence sum problem recap

- Given (possibly negative) integers A_1, A_2, …,A_n. Find the largest sum of a consecutive subsequence
- Divide and conquer approach:
  - One of the following must be true:
    - the max sum is in the first half of the sequence, or
    - the max sum is in the second half of the sequence, or
    - it begins in the first half and ends in the second half

Overview of algorithm

1. (recursively) Compute the max sum of first half of sequence
2. (recursively) Compute the max sum of second half of sequence
3. Compute (via two loops) the max of all sums that begin in the first half and end in the second half
4. Choose the largest of the three

```
/**
 * Recursive maximum contiguous subsequence sum algorithm.
 * Finds maximum sum in subarray spanning a[left..right].
 * Does not attempt to maintain actual best sequence.
 */
private static int maxSumRec( int [ ] a, int left, int right )
{
    int maxLeftBorderSum = 0, maxRightBorderSum = 0;
    int leftBorderSum = 0, rightBorderSum = 0;
    int center = ( left + right ) / 2;

    if( left == right )  // Base case
        return a[ left ] > 0 ? a[ left ] : 0;

    int maxLeftSum  = maxSumRec( a, left, center );
    int maxRightSum = maxSumRec( a, center + 1, right );

    for( int i = center; i >= left; i-- )
    {
        leftBorderSum += a[ i ];
        if( leftBorderSum > maxLeftBorderSum )
            maxLeftBorderSum = leftBorderSum;
    }

    for( int i = center + 1; i <= right; i++ )
    {
        rightBorderSum += a[ i ];
        if( rightBorderSum > maxRightBorderSum )
            maxRightBorderSum = rightBorderSum;
    }

    return max3( maxLeftSum, maxRightSum,
                 maxLeftBorderSum + maxRightBorderSum );
}
```

Analysis?

We need an approach to dealing with algorithms like this. Recurrence relations will help (see next slide).

Analysis?

- Let's do the special case where N is a power of 2
- Let T_N be the number of times we look at array elements if the array has size N, with $T_1 = 1$
- $T_N =$ (look back at the code)
- How do we solve this?

Another similar problem

- Given N points in the plane, find the two points that are closest to each other
- Running time for obvious algorithm
- Divide and conquer algorithm

The next step in analysis

- Recurrence relations
- These come up a lot in the analysis of recursive algorithms
- and sometimes iterative algorithms, also
- What is a recurrence relation?

Recurrence relation

- An equation (or inequality) that relates the n^{th} element of a sequence to certain of its predecessors
- Includes an initial condition
- Some of the techniques for solving them are reminiscent of techniques for solving differential equations
- When we solve a D.E., what do you get?
- When we solve a recurrence relation, what do you get?

Recurrence relation basics

- What does the solution of a recurrence relation describe?
  - a (usually infinite) sequence
  - how is this like solving a D.E.?
- What is the form of a recurrence relation?
  - initial value(s): value for c_0, or perhaps c_0, c_1, …c_k for some fixed number k.
  - a general formula that relates c_n to one or more earlier values in the sequence.
- Example: $g_0=0$, $g_n = 2 + g_{n-1}$.
- Example: $g_0=1$, $g_n = 2\ g_{n-1}$.
- Example: $f_0 = f_1 = 1$, $f_n = f_{n-2} + f_{n-1}$
- Example: $f_0 = 1$, $f_n = n*f_{n-1}$
- Example: $f_0 = 0$, $f_n = f_{n-1} + n$
- Example: $f_1 = 1$, $f_n = f_{n/2} + n$, where n is a power of 2.

With a partner, try to figure out the solutions to these.

Recurrence relations for in-class exercise

Student names:

----

1. Example: $f_0=0$, $f_n = 2 + f_{n-1}$.
   Solution: $f_n =$ _____
2. Example: $f_0=1$, $f_n = 2f_{n-1}$.
   Solution: $f_n =$ _____
3. Example: $f_0 = f_1 = 1$, $f_n = f_{n-2} + f_{n-1}$
   Solution: $f_n =$ _____
4. Example: $f_0 = 1$, $f_n = n*f_{n-1}$
   Solution: $f_n =$ _____
5. Example: $f_0 = 0$, $f_n = f_{n-1} + n$
   Solution: $f_n =$ _____
6. Example: $f_1 = 1$, $f_n = f_{n/2} + n$
   where n is a power of 2.
   Solution: $f_n =$ _____