Name: _____ Wisc id: _____

## Solving Recursion

1. Suppose you are choosing between the following three algorithms:

   - Algorithm A solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.
   - Algorithm B solves problems of size n by dividing them into nine subproblems of size $\frac{n}{3}$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time
   - Algorithm C solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.

   What are the running times of each of these algorithms (in big-O notation), and which would you choose?

**Solution:**

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

$B(n) = 9B(\frac{n}{3}) + n^2$

$C(n) = 5C(\frac{n}{2}) + n$

$T(n) = aT(\frac{n}{b}) + C(n) + D(n)$

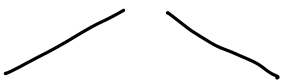
a = # of recursive calls

b = size of subproblem

C = runtime for combine

D = runtime for divide (usually 1)

## Bitonic Sequence

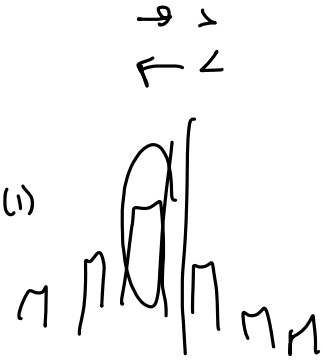


2. A bitonic sequence $x_1, ..., x_n$ is a sequence of numbers that is first non-decreasing then non-increasing. That is, $\exists j$ such that $\forall i < j$, $x_i \leq x_j$, and $\forall k > j$, $x_j \leq x_k$. Given a bitonic sequence, find the maximum element.

**Solution:**

given the sequence, split sequence in half, check the middle element, $\lfloor \frac{n}{2} \rfloor$, if the element $\lfloor \frac{n}{2} \rfloor + 1$ is $< \lfloor \frac{n}{2} \rfloor$ choose left half, if $\lfloor \frac{n}{2} \rfloor + 1$ is $> \lfloor \frac{n}{2} \rfloor$ choose right half. if sequence length == 1, that is the max element.

$T(n) = T(\frac{n}{2}) + O(1)$



## Power

3. Calculate $x^n$ in $\Theta(\log n)$ time.

**Solution:**

$$pow(x,n) = \begin{cases} x \cdot pow(x, \lfloor \frac{n}{2} \rfloor) \cdot pow(x, \lfloor \frac{n}{2} \rfloor), & \text{odd} \\ pow(x, \lfloor \frac{n}{2} \rfloor) \cdot pow(x, \lfloor \frac{n}{2} \rfloor), & \text{even} \end{cases}$$

$$T(n) = T(\tfrac{n}{2}) + O(1)$$

## Fake Coin

4. You are given n coins, where $n = 3^k$. They all look identical. They should all be the same weight, too – but one is a fake, made of a lighter metal.

   Your neighbor has an old-fashioned balance scale that enables you to compare any two sets of coins. If it tips either to the left or to the right, you will know that the one of the sets is heavier than the other. Sadly, you aren't on speaking terms with the neighbor, so he charges you each time you weigh anything.

   Design an algorithm to find the fake coin in the fewest number of weighings.

**Solution:**

since $n = 3^k$, we should divide the set by 3 groups and compare 2 of them. if the scale doesn't move the 3rd set has OG coin if it moves left or right the one that is lighter has the OG coin. The stack w/ OG coin continue process of divide by 3.

# Convex Hull

5. Given $n$ points on the $x - y$ plane, determine the smallest convex polygon that contains all the points.

**Algorithm 1** `LOWER-TANGENT`($A$, $B$)

```
Let a be the rightmost point in A
Let b be the rightmost point in B
while ab is not the lower tangent to A and B do
    while ab is not the lower tangent to A do
        move a clockwise
    end while
    while ab is not the lower tangent to B do
        move b counter-clockwise
    end while
end while
```

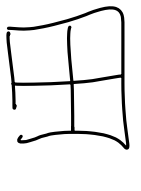
**Solution:**

## Tromino Tiling

6. Given a $n$ by $n$ board where $n = 2^k$ and $k \geq 1$. The board has one missing cell (of size $1 \times 1$). Fill the board using L shaped tiles. A L shaped tile is a $2 \times 2$ square with one cell of size $1 \times 1$ missing.

**Solution:**

divide board in 4 sections keep doing to find missing cell. Once missing cell found place L shaped square in center to cover all other sections so each has an extra space. now fill the board.




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