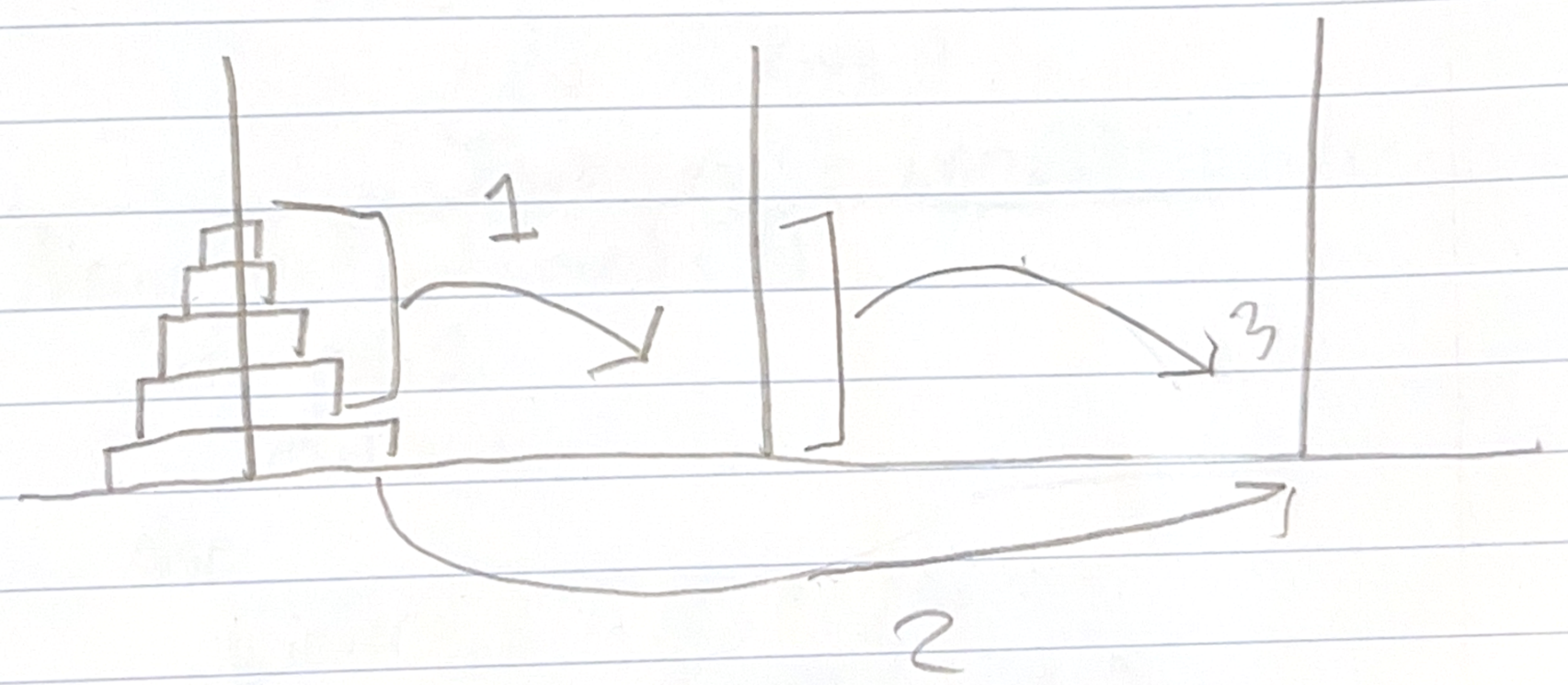
Oct 21 Exam Review Notes

Tower of Hanoi:

```
Solve (n, Start, Dest, Spare)
    if n = 1
        move the disk from start to dest
    else
1       Solve(n-1, start, spare, dest)
2       move largest disk to dest
3       Solve(n-1, spare, dest, start)
```



$T(1) = 1$

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

Oct 21 CS375 Exam Review Notes

Inversion Counting

Given an array A, how many pairs $i<j$ where $A[i] > A[j]$

```
MergeandCount(L,R)
    Output = []
    Lindex = 0
    Rindex = 0
    while(len(output) < len(L) + len(R))
        if Rindex = len(R) or L[Lindex] ≤ R[Rindex]
            add L[Lindex] to out
        else
            add R[Rindex] to out
            Inversions += Len(L) - Lindex
```

```
MergeSortandCount(A)
    if n≤1
        return 0
    else
        Linverse, C = MSAC(A[:n/2])
        Rinverse, C2 = MSAC(A[n/2:])
        (A, Minverse) = MAC(L,R
            return (A, C+C2+Minverse)
```

Oct 21 CS375 Exam Review Notes

Master Theorem:

Let $T(n) = aT(n/b) + f(n)$

Let $c^* = \log_b a$

If $f(n) = O(n^c)$ for some $c < c^*$

$T(n) = \Theta(n^{c^*})$

If $f(n) = \Theta(n^{c^*} \log^k n)$

$T(n) = \Theta(n^{c^*} \log^{k+1} n)$

If $f(n) = \Omega(n^c)$ for some $c > c^*$

$T(n) = \Theta(f(n))$

Ex: $T(n) = 21T(n/7) + n^2$

$\log_{21} 7 = c^* = 1.? < 2$

Case 3 $\rightarrow T(n) = \Theta(n^2)$

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

$c^* = \log_{3/2} 8 > 2$

notice reciprical ↗ Case 1 $\rightarrow T(n) = \Theta(n^{\log_{3/2} 8})$

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

$\sum_{i=0}^{\lg n - 1} \frac{n}{\lg(n/2^i)} = n \sum \frac{1}{\lg n - i}$

$= n \sum_{i=1}^{\lg n - 1} \frac{1}{i} = \Theta(n \lg \lg n)$

