6/18/25

1. Tests back + Review
2. Towers of Hanoi
3. Binary Search
4. Math for comp. prog (number theory)

Towers (n=5 disks, s=1, e=2)

1. Towers(n=4 disk, s=1, e=3)
2. Move disk n=5 from s=1 to e=2
3. Towers (n=4, s=3, e=2)

```
towers(n, s, e) {
    towers (n-1, s, mid)
    move (n, s, e)
    towers (n-1, mid, e)
}
```

Let $T(n)$ = # moves w/ n disks

$T(1) = 1$

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

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

w/ rec rel

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

Binary Search

low high

arr

3	6	12	14	90	101	125	180	191	202
0	1	2	3	4	5	6	7	8	9

val $\boxed{180}$ Y

$\boxed{6}$ Y

$\boxed{121}$ N

```
bool binseach(vector<int>& arr, int low, int high,
                                   int val){

    if  (low > high) return false
    int  mid = (low + high) / 2;
    if  ( val < arr[mid])
        return binsearch(arr, low, mid-1);
}   else if (val > arr[mid] )
        return binsearch(arr, mid+1, high)

    else
        return true;
```

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

$$n = 2^k$$

$$\boxed{k = \log_2 n}$$

Prime Test

```
bool isprime (int n) {
    if (n<2) return false;
    for (int i=2; i<n; i++)
        if (n%i == 0)
            return false;
    return;
}
```

Runs n steps $O(n)$.

$195 = \underline{13} \times \underline{15}$

if n is composite, it has at least one divisor, (not prime > 1)

$a, 1 < a \leq \sqrt{n}$.

Assume n is composite and has no divisor $\leq \sqrt{n}$

$n = ab$ and $a > \sqrt{n}$ and $b > \sqrt{n}$

$> \sqrt{n} \times \sqrt{n}$

$= n \rightarrow$ Contradiction $n > n$.

In general most times when $ab = n$, one # is $< \sqrt{n}$ and one # is $> \sqrt{n}$.

exception perfect square $169 = 13 \times 13$

Improved Prime Test

// if n is long long i has to be also.

```
bool isPrime(int n) {
    if (n<2) return false;
    for (int i=2; i*i <= n; i++)
        if (n % i == 0)
            return false;
    return true;
}
```

$O(\sqrt{n})$ OK upto about $n \approx 10^{12}$

Prime Sieve (Sieve of Eratosthenes)

~~1~~	2	3	~~4~~	5	V
~~6~~	7	~~8~~	~~9~~	~~10~~	
11	~~12~~	13	~~14~~	~~15~~	
~~16~~	17	~~18~~	19	~~20~~	
~~21~~	~~22~~	23	~~24~~	~~25~~	

Implementation
Boolean Array
set True
CROSS OFF → set to false

Prime Factorization

$96 = 2^5 \times 3$

$75 = 3 \times 5^2$

Prime Test

→ trial division upto square root
when we find a divisor we stop
INSTEAD → count how many times a
prime divides in

96	i		pair
96	2	$96 \xrightarrow{+1} 48 \xrightarrow{+1} 24 \xrightarrow{+1} 12 \xrightarrow{+1} 6 \xrightarrow{+1} 3$	$\boxed{2^5}$
↓ 3	3	$3 \xrightarrow{+1} 1$	$\boxed{3^1}$

Go through potential divisors, if one divides in count how many times, divide this out of the #, store this term!

Factorial Power Problem

$m!$ how many times does n divide in?

$n = p_1^{a_1} p_2^{a_2} p_3^{a_3}$ ① Prime Factorize n.

$n = 2^5 \times 3$ $\qquad m = 100$

$$\begin{array}{r|l} 2 & 100 \\ 2 & 50 \\ 2 & 25 \\ 2 & 12 \\ 2 & 6 \\ 2 & 3 \\ & 1 \end{array}$$

$$\begin{array}{r} 50 \\ 25 \\ 12 \\ 6 \\ 3 \\ 1 \\ \hline 97 \end{array}$$

$$\left\lfloor \frac{97}{5} \right\rfloor = 19$$

$$\begin{array}{r|l} 3 & 100 \\ 3 & 33 \\ 3 & 11 \\ 3 & 3 \\ & 1 \end{array}$$

$$\begin{array}{r} 33 \\ 11 \\ 3 \\ 1 \\ \hline 48 \end{array}$$

for each prime factor p^a calculate # times p divides into $n!$

Call this X

Compute $\frac{X}{a}$

min all of these