Recurrence Relations

- A recurrence relation defines a function $T : \{0,1,2,\ldots\} \rightarrow \mathbb{R}$ using a recursive rule

- $T(n)$ is defined in terms of $T(0), T(1), T(2), \ldots, T(n-1)$

- Enough base cases (eg. $T(0)$, $T(1)$) are explicitly given so that $T(n)$ is defined, recursively when not defined explicitly.

Ex:

$$T(n) = \begin{cases} 0 & n=0 \\ T(n-1)+n & n \geq 1 \end{cases}$$

n	0	1	2	3	4	5
$T(n)$	0	1	3	6	10	15

- Recurrence relations often show up when we analyze [redacted] recursive algorithms.

```
Sort(A[1..n]):
    if n=0 return;
    max_index ← 1
    for i=1 to n do
(*)     if A[i] > A[max_index] then
            max_index ← i
    Swap(A, max_index, n)
    Sort(A[1...n-1])
```

- What is the runtime of Sort(A[1..n])?
- This is roughly the number of times the line (*) is executed.
- Define $T(n)$ to be the number of times (*) is executed throughout the execution of Sort(A[1..n]).

- Note that $T(n)$ satisfies the recurrence relation

$$T(n) = \begin{cases} 0 & n = 0 \\ n + T(n-1) & n \geq 1 \end{cases}$$

- How big is $T(n)$? How do we find ~~evaluate~~ ~~T(n)?~~ a simple formula for $T(n)$?

- Not always easy. We'll learn a few tricks which will help us attack the kinds of recurrence relations that show up in algorithm analysis.

- The first step to understanding what is going on is to compute some small values.

- Technique: Guess and Check. If you can guess the correct solution, it is usually easy to check your guess with induction.

n	0	1	2	3	4	5	6
$T(n)$	0	1	3	6	10	15	21

- Perhaps these numbers look familiar from Pascal's Triangle (Lecture 2):

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

⋮

- Because these numbers appear in some row m and are in the column position $k=2$ (remember, indexing starts at $k=0$), these numbers all have the form $\binom{m}{2}$ for some integer m.

- To get the sequence to match up properly, we $\boxed{\text{guess}}$ $T(n) = \binom{n+1}{2}$.

- Next, we must $\boxed{\text{check}}$ our answer with an inductive proof.

<u>Prop</u> For each $n \geq 0$, $T(n) = \binom{n+1}{2}$.

<u>Pf</u>: By induction on n.

If $n=0$, then $T(n)=0$ and $\binom{1}{2}=0$.

If $n \geq 1$, then $T(n) = T(n-1) + n$. By the inductive hypothesis, $T(n-1) = \binom{n}{2}$.

Therefore

$$
\begin{aligned}
T(n) &= T(n-1) + n \\
&= \binom{n}{2} + n \\
&= \binom{n}{2} + \binom{n}{1} \\
&= \binom{n+1}{2}
\end{aligned}
$$

where the last equality is a special case of a Theorem in Lecture 2 (just before Pascal's Triangle.) ∎

- So how big is $T(n)$?

$$\frac{1}{2}n^2 \leq T(n) = \binom{n+1}{2} = \frac{n(n+1)}{2} \leq n^2$$

- So: $\frac{1}{2}n^2 \leq T(n) \leq n^2$

- Later, we will ~~say~~ write $T(n) = \Theta(n^2)$.

<u>Remark</u>: The checking step is <u>crucial</u>.

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n-1) + \binom{n-1}{3} + n - 1 & n \geq 2 \end{cases}$$

n	1	2	3	4	5	...
$T(n)$	1	2	4	8	16	

It is natural to guess $T(n) = 2^{n-1}$ is always a power of two.

$\boxed{\text{But}}$: $T(6) = 31$

- If we tried to prove $T(n) = 2^{n-1}$ inductively, we would see that our guess is not correct.

<u>Remark</u>: $T(n) = \binom{n}{4} + \binom{n}{2} + 1$.

7.5

$T(1) = 1$

$T(2) = T(2-1) + \binom{2-1}{3} + 2-1 = T(1) + \binom{1}{3} + 1 = 2$

$T(3) = T(2) + \binom{3-1}{3} + 2 = 2 + 0 + 2 = 4$

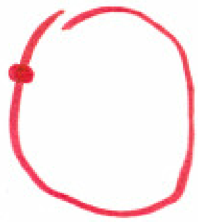
$T(4) = T(3) + \binom{3}{3} + 3 = 4 + 1 + 3 = 8$

$T(5) = T(4) + \binom{4}{3} + 4 = 8 + 4 + 4 = 16$

!!! $T(6) = T(5) + \binom{5}{3} + 5 = 16 + 10 + 5 = 31$

$= \frac{5!}{3! \cdot 2!}$

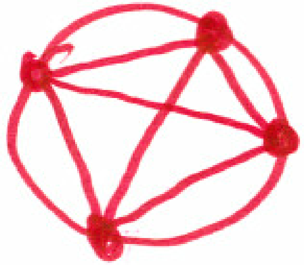
Also, $T(n)$ is the maximum number of regions inside a circle with n points on the circumference and lines drawn between the points.



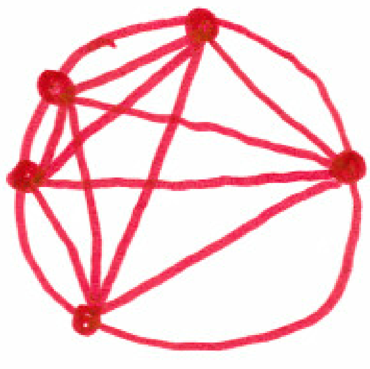
$n=1$
$T(n)=1$

$n=2$
$T(n)=2$

$n=3$
$T(n)=4$



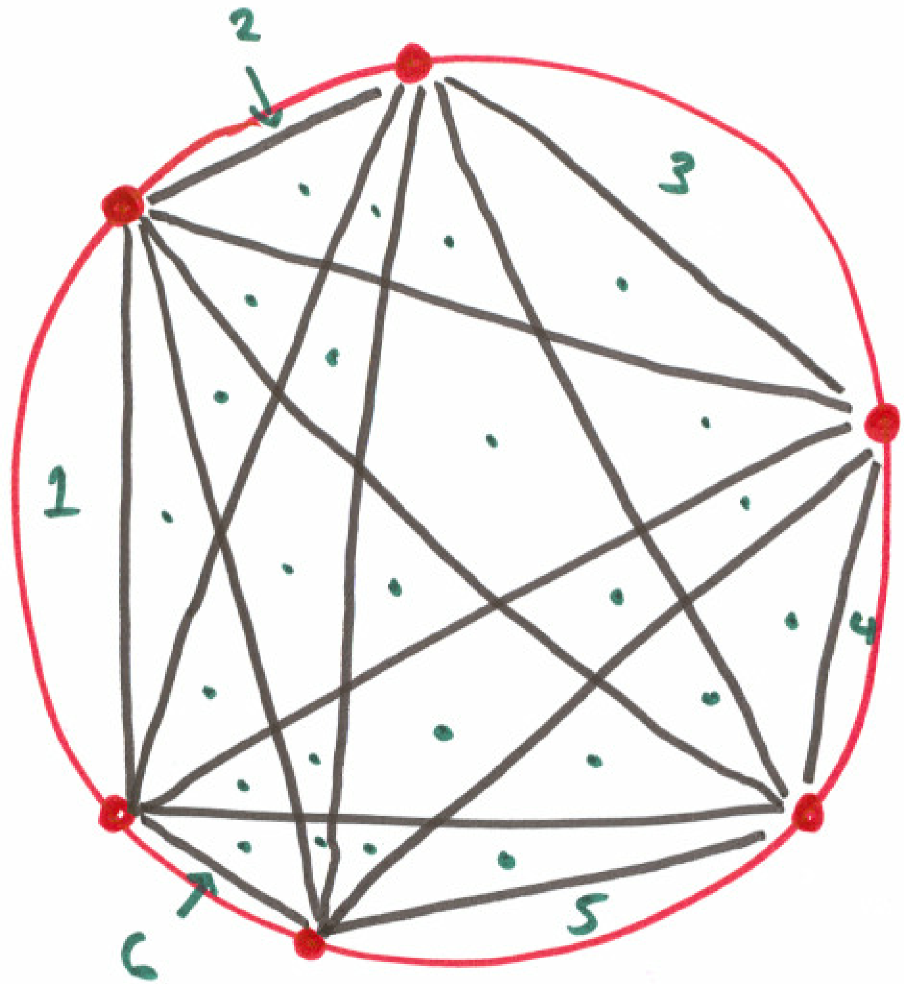
$n=4$
$T(n)=8$



$n=5$
$T(n)=16$

$n=6$

$T(6)=31$



Another Technique: Unroll the recursion.

$$T(n) = \begin{cases} 0 & n = 0 \\ T(n-1) + n & n \geq 1 \end{cases}$$

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

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

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

$$= \cdots$$

$$= 0 + 1 + 2 + \cdots + n$$

So $T(n) = \sum_{j=1}^{n} j = \binom{n+1}{2}$.

If we can unroll the recursion and obtain a sum, that is progress:

Sums are easier to solve than recurrences.

- What if we figure out $T(n) = \sum_{j=1}^{n} j$ but we forget $\sum_{j=1}^{n} j = \binom{n+1}{2}$?

- Here is a useful trick to estimate sums; suppose n is even:

$$1 + 2 + \cdots + \frac{n}{2} + \cdots + n \leq \underbrace{n + n + \cdots + n}_{n \text{ terms}} = n^2$$

$$\underbrace{1 + 2 + \cdots + \frac{n}{2}}_{n/2 \text{ terms}} + \underbrace{\cdots + n}_{n/2 \text{ terms}} \geq \underbrace{0 + 0 + \cdots + 0}_{n/2 \text{ terms}} + \underbrace{\frac{n}{2} + \frac{n}{2} + \cdots + \frac{n}{2}}_{n/2 \text{ terms}}$$

$$= \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$$

- So even without solving the sum, we know

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

and can say $T(n) = \Theta(n^2)$.