Lecture 11

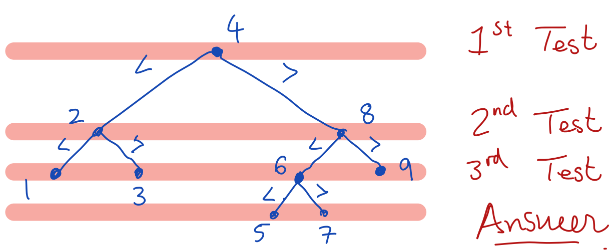
Eg. How many binary tests to find a number? $\in \{1, \dots, N\}$.

This describes the following situation: You have a number in a box ▨ you can't see, but you know $1 \leq ▨ \leq N$.

You have an operation (▨, 5) which tests box against a number (in this case 5) and returns ▨ < 5, ▨ $= 5$, ▨ > 5.

We model this w/ a tree. Each vertex represents a test (▨, ?) and each edge represents either a $<$ or a $>$ outcome:

Eg: $N = 9$:



Every number appears in the tree so if $1 \leq ▨ \leq 9$ then somewhere in the tree will be a vertex "test" which tells you what ▨ is equal to.

These tests are called binary because they make a binary tree!

This search tree is <u>balanced</u> b/c all the leaves occur in the bottom two levels.

In general, a balanced, binary tree is the fastest way to search with this type of test.

<u>Q</u>: How many tests do we need? <u>Ans</u>: We only need one test at each level, b/c each ▨ only travels down one branch.

<u>Q</u>: So how many levels?

<u>Soln</u>: for a binary tree $m = 2$, and N vertices total we get

$$\left\{ \begin{array}{l} n = \ell + i \\ n = mi + 1 \end{array} \right\} \quad \& \quad h = \lceil \log_m(\ell) \rceil$$

where h = height = # of levels.

True for any m-ary tree

True for balanced m-ary trees

$m = 2$, so we need ℓ to get h.

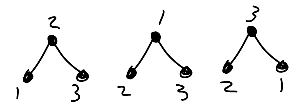
$$\begin{array}{l} N = \ell + i \\ N = 2i + 1 \end{array} \Rightarrow N - \ell = \tfrac{1}{2}(N-1) \Rightarrow 2N - 2\ell = N - 1$$

$$\Rightarrow N = 2\ell - 1 \Rightarrow \ell = \frac{N+1}{2} \Rightarrow \boxed{h = \lceil \log_2(N+1) \rceil - 1}$$

Q: How many search trees could we make?

A search tree is really a tree with $\mathbb{Z}$-valued labels so

Thm [Cayley] There are N^{N-2} non-isomorphic labeled trees with labels $1 \dots N$.

Eg $N = 3 \Rightarrow N^{N-2} = 3^{3-2} = 3$. and these are: 

Any other labeled tree with 3 labels is isomorphic to one of these.

Pf: We are interested in the size of the set of N-labeled trees. so we will define a new set $\mathcal{S}$ which contains exactly one sequence of numbers for each N-labeled tree. We will be able to count the sequences easily, and this will be the number of trees.

Let $\mathcal{S}$ be sequences of the numbers $1, \dots, N$ with length $N-2$.

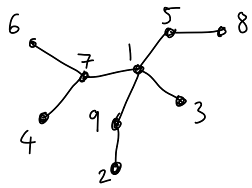
Eg for $N=4$, $\mathcal{S} = \{ (1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), (3,1), (3,2), (3,3), (3,4), (4,1), (4,2), (4,3), (4,4) \}$ All sequences of length $4-2=2$.

* How many are there? $(\underset{1}{\_}, \underset{2}{\_}, \underset{3}{\_}, \dots, \underset{N-2}{\_})$ There are $N-2$ places and N choices for each place so $\underset{1}{N} \cdot \underset{2}{N} \cdot \dots \cdot \underset{N-2}{N} = N^{N-2}$ choices! Thus $|\mathcal{S}| = N^{N-2}$.

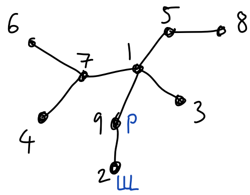
But Why is $|\mathcal{S}| = \#$ of N-labeled graphs?

B/c there is a process for turning a tree into a sequence and another for turning a sequence to a tree.

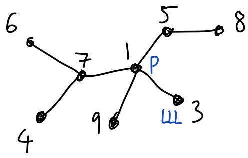
Turning a tree to a sequence:



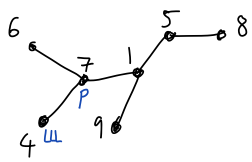
Process: Find Lowest Labeled Leaf "L.L.L."
Find its Parent
Add the parent to the list
Delete the leaf
Repeat
Stop when 2 vertices remain.

① 

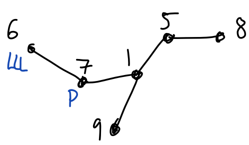
L.L.L. = 2, parent = 9 so List = (9,

② 

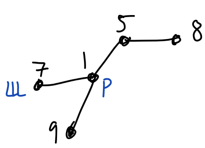
L.L.L. = 3 parent = 1 so List = (9, 1

③ 

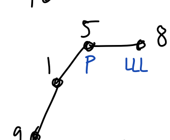
L.L.L. = 4 parent = 7 so List = (9, 1, 7

④ 

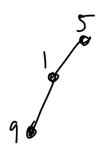
L.L.L. = 6 parent = 7 so List = (9, 1, 7, 7

⑤ 

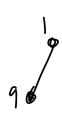
L.L.L. = 7 parent = 1 so List = (9, 1, 7, 7, 1

⑥ 

L.L.L. = 8 parent = 5 so List = (9, 1, 7, 7, 1, 5

⑦ 

L.L.L. = 5 parent = 1 so List = (9, 1, 7, 7, 1, 5, 1

⑧ 

Two left so STOP, list = $(9,1,7,7,1,5,1) \in \mathcal{S}$.

$N-2 = 7$ entries

Turning a Sequence into a tree: Eg (5, 2, 2, 3, 3, 3).

Write [1, ..., N] below the sequence then connect a vertex labeled with the smallest available number in [...], which is NOT left in the sequence, to the vertex with the left most entry of the sequence then remove that entry from the sequence and [...]s.

Eg: ⓪ [1, 2, 3, 4, 5, 6, 7, 8] Setup.
(5, 2, 2, 3, 3, 3)

① [1, 2, 3, 4, 5, 6, 7, 8]
(5, 2, 2, 3, 3, 3) → No 1's
leftmost.

① [~~1~~, 2, 3, 4, 5, 6, 7, 8]
(~~5~~, 2, 2, 3, 3, 3) → No 4's
leftmost.

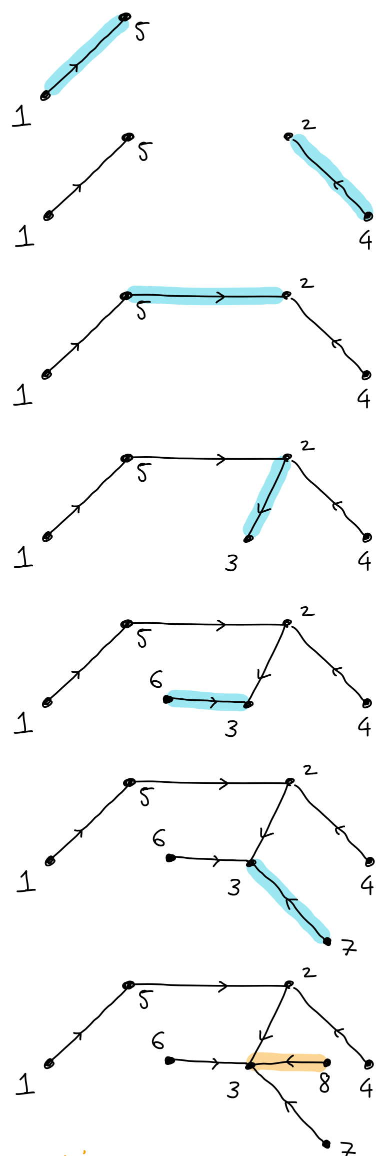
① [~~1~~, 2, 3, ~~4~~, 5, 6, 7, 8]
(~~5~~, ~~2~~, 2, 3, 3, 3) → No 5's
leftmost.

① [~~1~~, 2, 3, ~~4~~, ~~5~~, 6, 7, 8]
(~~5~~, ~~2~~, ~~2~~, 3, 3, 3) → No 2's
leftmost.

① [~~1~~, ~~2~~, 3, ~~4~~, ~~5~~, 6, 7, 8]
(~~5~~, ~~2~~, ~~2~~, ~~3~~, 3, 3) → No 6's
leftmost.

① [~~1~~, ~~2~~, 3, ~~4~~, ~~5~~, ~~6~~, 7, 8]
(~~5~~, ~~2~~, ~~2~~, ~~3~~, ~~3~~, 3) → No 7's
leftmost.

① [~~1~~, ~~2~~, 3, ~~4~~, ~~5~~, ~~6~~, ~~7~~, 8]
(~~5~~, ~~2~~, ~~2~~, ~~3~~, ~~3~~, ~~3~~)



Finish by connecting last two vertices.

If you can turn graph → sequence and sequence → graph then there are the same number of each. There are N^{N-2} sequences so there are N^{N-2} labeled trees