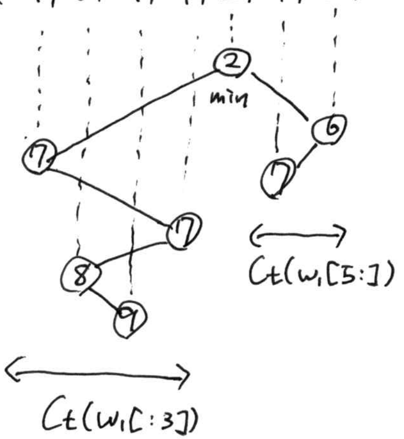
Cartesian trees. (Vuillemin '80)

- Consider a string w over the Integer alphabet.
- A Cartesian tree $Ct(w)$ of w is recursively defined as:
  - > if $|w| = 0$, $Ct(w)$ is empty.
  - > o/w, let i be the smallest idx with the minimum $w[i]$.
    - The root of $Ct(w)$ is labeled $w[i]$
    - The left subtree of the root is $Ct(w[:i-1])$
    - The right subtree ———"——— $Ct(w[i+1:])$
- Two Cartesian trees are <u>equivalent</u> if they have the same structure.

<u>Example</u> $w_1 = (7, 8, 9, 7, 2, 7, 6)$



$w_2 = (2, 8, 16, 4, 1, 3, 2) \Rightarrow Ct(w_1) = Ct(w_2)$

Known facts.

- $Ct(w)$ can be constructed in $O(|w|)$ time. (Gabow et al. '84)
- $Ct(w_1) \overset{?}{=} Ct(w_2)$ can be tested in $O(|w_1| + |w_2|)$ time.
- The pattern matching problem w.r.t. Cartesian trees can be ~~sto~~ ~~sto~~ solved in $O(|T| + |P|)$ time. (Park et al. '19)

Problem. [Approximate Cartesian tree pattern matching]

(Informal) Let the Cartesian edit distance from string u to string v be the min. total cost of edits <u>on u</u> to make another string u' allowing $Ct(u') = Ct(v)$.

Denote as $Cdist(u \to v)$.

Given a text T, ~~and~~ a pattern P and a threshold t, compute all substrings w of T that satisfy

<u>$Cdist(w \to P) \le t$.</u>

↳ ☆ <u>Compute this</u> ☆

The Cartesian edit distance is asymmetric.

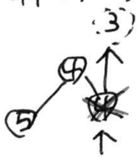
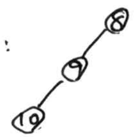
Consider $w = (3, 4, 5, 5, 4, 3)$ and $u = (4, 5, 5, 5, 4)$.

$Cdist(w \to u) = 1$ but $Cdist(u \to w) = 2$.

(assume unit cost for insert / delete / substitute.)

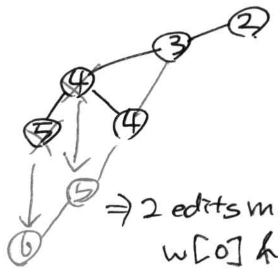
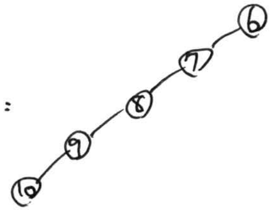
Greedy does not work.

Consider $w = (5,4,4,3,2)$ and $u = (10,9,8,7,6)$

$Ct(w[:2])$:  and $Ct(u[:3])$: 

$\Rightarrow$ 1 edit is enough

However,

$Ct(w)$:  and $Ct(u)$: 

$\Rightarrow$ 2 edits made on $w[0]$ & $w[1]$

Observe the root value of the "edited" Cartesian tree may differ ~~according to what edits~~ between edit sequences. This may disallow some future edit sequences if the root value is fixed ~~to~~ w.r.t. the current best ~~version of edits~~ edit sequence.

Naive upper-bound on Cdist$(w \to u)$

Consider deleting all chars in w except one, and inserting the $|u|-1$ other chars to make the same Cartesian tree.

$\Rightarrow$ Cdist$(w \to u)$ is $O(|w|+|u|)$.

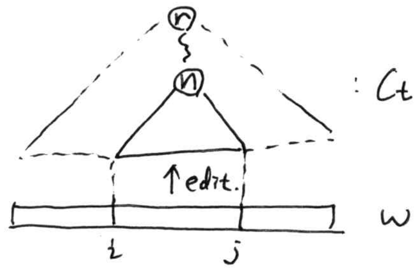
Idea for computing Cdist$(w \to u)$

> Consider every 3-tuple (i, j, n):

- $0 \le i \le j \le |w|$ and n is a node in $Ct(u)$

~~(Conceptually)~~

Call each 3-tuple a subproblem; edit $w[i:j-1]$ to match the subtree rooted at n in $Ct(u)$



Recall that more edits in this subproblem may lead to higher root labels in the edited Ct.

$\Rightarrow$ Save (max root label, edit cost) tuples for all edit costs under the naive upper bound for Cdist$(w \to u)$.

i.e., we can define a function that maps (i, j, n, x) to a max root label.

(x $\to$ target edit cost.)

Call that function opt.

Base cases. ~~Consider~~ Let n be a leaf node.

1. $opt(i, i, n, 1) = \infty$ (1 insert)
2. $opt(i, j, n, j-i-1) = \max\{w[k] \mid i \le k \le j-1\}$
3. $opt(i, j, n, j-i) = \infty$ ($j-i-1$ deletes, 1 substitution)
4. $opt(i, j, n, x) = -\infty$ forall undefined x's.
   ↳ "impossible"

Recurrence. Let n be an internal node.

W.l.o.g. let n have two children (l, r) by placing dummy nodes n' satisfying $opt(i, i, n', 0) = \infty$ and $opt(i, j, n', x) = -\infty$ o/w.

1. $insOpt(i, j, n, x, a) = \max\limits_{\substack{y, z : y, z \ge 0 \\ y+z = x-1}} \big(\min(opt(i, a, l, y) - 1,\ opt(a, j, r, z))\big)$

2. $subOpt(i, j, n, x, a) = \max \begin{cases} \max\limits_{\substack{y, z : y, z \ge 0 \\ y+z = x-1}} \big(\min(opt(i, a, l, y) - 1,\ opt(a+1, j, r, z))\big) \\ w[a] \quad \text{if } w[a] \le \max\limits_{\substack{y, z : y, z \ge 0 \\ y+z = x}} \big(\min(opt(i, a, l, y) - 1,\ opt(a+1, j, r, z))\big) \end{cases}$

3. $opt(i, j, n, x) = \max\limits_{a \in [i, j]} \big(\max(insOpt(i, j, n, x, a),\ subOpt(i, j, n, x, a))\big)$

⇒ Computing $opt(0, |w|, root, x)$ for all possible x's take $O(|w|^3 |u| D^2)$ time (D is the upper bound for $(dist(w \to u)$.)

Problem solution.

> Note: $opt(i, j, root, x)$'s are all computed through a single run of the recurrence.

⇒ Run the alg that computes $\underline{opt}$, then return all (i, j) pairs with x that satisfy $opt(i, j, root, x) \le t$.

If t is constant, we have $D \le t$; therefore the overall RT is $O(|T|^3 |P|)$.