3-sided emptiness

use Cartesian tree

$O(n)$ space, $O((\log \log U)^3)$ time

by 2 pred/succ searches + 1 LCA query

general 4-sided:

use range tree

store 3-sided DS at each node

$O(n \log n)$ space

$O((\log \log U)^3)$ time by 2 3-sided queries

($+$ for reporting)

Advanced Method 2: Alstrup, Brodal, Rauhe '00

$O(n(\log \log n))$ space, $O((\log \log U)^3)$ time

Idea: $\sqrt{n}$-way recursion

form a grid with $\sqrt{n} \times \sqrt{n}$ cells

where each row/column has $\sqrt{cn}$ pts
recursing in each row & in each column
space $O(n)$

store 3-sided DS in each row & column

store list $L$ of all nonempty grid cells in 4-sided DS

- space $O(141 \log 141) = O(\frac{4}{3} \log \frac{4}{3})$
  
  $= O(n)$ by setting $C = \log n$

Analysis of space:

let $s(n) =$ space per pt

$s(n) \leq 2 s(\sqrt{Cn}) + O(1)$

pretend $C=1$

$\Rightarrow \quad s(n) = O(2 \log \log n) = O(\log n)$ bad!

idea - bit packing + "rank space reduction"

( replace coords by ranks, U-in)

$s(n) \leq 2 s(\sqrt{Cn}) + O(\log \frac{n}{w}) \log n$

pretend $C=1$

$\Rightarrow \quad s(n) = O(\log \frac{n}{w} + 2 \log \sqrt{n} + 4 \log \frac{n^{1/4}}{w} + ...)$

$= O(\log \frac{n}{w} + \log \frac{n}{w} + \log \frac{n}{w} + ...)$

$= O(\log n \log \log n)$

$\Rightarrow \quad S(n) = O(n \log \log n)$
Querying (emptiness)

\[ Q(n) \leq \max \{ Q(\sqrt{n}) + O(\log \log U), \quad O(\log \log U) \} \]

+ 3-sided queries
+ 1 4-sided query in L

\[ \Rightarrow Q(n) = O(\log \log U) \]
(+ k log log U for report)

Advanced Method 3: C. Larsen-Patrascu '11

Idea - go back to range tree!
- Save space by bit packing + succinct DS
  (Succinct rank DS)

Fact 1: can store array A of n symbols in \([0]^n\)
using \(O(\frac{n \log \sigma}{w})\) words of space

s.t. can compute \(\text{rank}_A(i) = \# \) times \(A[j] \) occurs
in \(A[l \ldots i]\)
in \(O(1)\) time

Pf sketch: divide into blocks of size \(\sigma w\)

\[ \sigma w \quad \sigma w \quad \sigma w \]

store \(\sigma\) counts per block

\[ \Rightarrow \text{space} O(\frac{n}{\sigma w} + \sigma) = O(n/w) \text{ words} \]
store answers within each block

\[ \text{space } O\left( \frac{n}{\sigma w}, \sigma w \cdot \log(\sigma w) \right) = O\left( \frac{n \log(\sigma w)}{w} \right) \]

can get rid of \( \log w \) by more blocking.

\[ \text{8th position} \]

\[ 130032321012 \]

\[ \text{rank}_2(8) = 2 \]

\[ \text{8th lowest pt} \]

\[ = \text{2nd lowest pt in slab2} \]

**Key Lemma**: can compress 2D range tree in \( O(\log \log n) \) space

5th. Given any node \( v \) & index \( i \), can look up \( i \)th element in \( v \)'s y-sorted list in \( O(\log \log n) \) time.

**Pf**: Use Fact 1 repeatedly.
query: follow $O(\log \log n)$ "pointers" (by rank op)

\[
\text{space: } O\left( \frac{n \log \log n}{w} \log n + \frac{n \log \log n}{w} + \frac{n \log \log n}{w} \right)
\]

\[
= O\left( \frac{n \log \log n}{w} \log n \right) \leq O(n \log \log n).
\]

(succinct LCA)

Fact 2: can store any binary tree in $O(w)$ words of space

st. can answer LCA query in $O(1)$ time

(succinct pred)

Fact 3: given sorted list $y_1, \ldots, y_n$ in $\mathbb{U}$,

can build DS in $O\left( \frac{n}{w} \log w \right)$ words of space

st. pred search takes $O(\log \log \mathbb{U})$ time

+ $O(1)$ lookups of the $y_i$'s.

Pf Sketch: select subset containing every other $b$ elements

build VEB tree for subset $b = w / \log w$

for each block of $b$ elements

build compressed trie (recall fusion tree)

\[
O\left( \frac{n}{b} \cdot \log w \right) \text{ space}
\]

Query algm for 4-sided:

$\Rightarrow$ 2 3-sided queries

$\Rightarrow$ 2 pred searches by Fact 3

(concat lookup by Key Lemma)

$\Rightarrow$ $O(\log \log \mathbb{U})$ time

(+ $k \log \log n$ for reporting)

Space: $O\left( \frac{n \log \log n}{w} + \frac{n \log \log n}{w} \log n \right) = O(n \log \log n)$

Key Lemma

Fact 2 & 3 for each level

For $w = \log n$
Remark: trade-offs

$O(n \log^3 n)$ space, $O(\log \log U + k)$ time

OR $O(n)$ space, $O(\log^2 U + k \log^3 n)$ time

Remark: 3D dominance (3-sided) emptiness

Decide if a box staircasepolyhedron

$O(n)$ space

$O(\log \log U)$ time
(as we'll see ...)

3D general (6-sided):

$O(n \log^3 n)$ space, $O(\log \log U)$ time

Improves to

$O(n \log^{1+\varepsilon} n)$ space, $O(\log \log U)$ time
(+ k for report)

[C. - Larsen - Patrascu '11]
[using recursive grids]

4D: use b-ary range tree

Set $b = \log^2 n$

$O(\frac{n \log^{1+\varepsilon} n \cdot b \log n}{\log b})$ space $\Rightarrow$ $O(n \log^{2+\varepsilon} n)$ space

$O(\frac{\log \log U \cdot \log n}{\log b})$ time $\Rightarrow$ $O(\log n)$ time

[Patrascu: lower bound $\Omega \left( \frac{\log n}{\log \log n} \right)$]