COMP4128 Week 03 Tutorial

Yifan He

z5173587@unsw.edu.au
https://github.com/hharryyf/COMP4128-24T3-tutoring

Outline

- Data structure recap
- Ghost Encounters
- Restructuring Company
- Problem set 2 hints by email

Data structure recap

C++ STL

Data structure	Operation	Complexity
vector	push_back pop_back access i-th element	$O(1)$
stack	push/pop/top	$O(1)$
queue	push/pop/front	$O(1)$
priority_queue	top pop/push	$O(1)$ $O(log_2 N)$

Data structure recap (cont.)

set/map

- insert/erase/find
- lower_bound(v): an iterator to the smallest element $\geq v$
  - To get the largest element $< v$, use $*prev(it)$
- upper_bound(v): an iterator *it* to the smallest element $> v$
  - To get the largest element $\leq v$, use $*prev(it)$
- All elements above works in $O(log_2 N)$ time

Data structure recap (cont.)

k-th order statistics

- insert/erase/find
- find_by_order(k): get the k-th smallest element (0-index)
- order_of_key(v): get the order of v (0-index)
- All the above operations are in $O(log_2 N)$ time

Data structure recap (cont.)

Union find

- Given N vertices, no edge
- Connect u and v with an undirected edge
- Query if x and y are reachable
- $O(1)$ per operation
- Union find does not support arbitrary edge deletion

Data structure recap (cont.)

Range tree

- Given an array a of size N
  - Query the min/max/sum of elements of $a[L, R]$
    - Not just min/max/sum, can be other divide and conquer properties
    - V has two children L and R, V can be computed easily with information stored in L and R
  - Update the i-th value of the array to v
  - Update all elements $a[L, R]$ to v (later weeks)
  - Increment all elements $a[L, R]$ by v (later weeks)
- $O(log_2 N)$ per operation

Ghost Encounters

There are N ($N \leq 100,000$) ghosts, each is going to appear at position X_i at time T_i seconds. A person starts moving at time S seconds and position 0. For each unit distance, the person needs to use K seconds. (Note that S can be negative). By picking the optimal S, what is the maximum number of ghosts the person can encounter?

Ghost Encounters

Analysis

- The speed of the person is $\frac{1}{K}$.

Ghost Encounters

Analysis

- The speed of the person is $\frac{1}{K}$.
- What is the condition of S that the person can encounter the i-th ghost?

Ghost Encounters

Analysis

- The speed of the person is $\frac{1}{K}$.
- What is the condition of S that the person can encounter the i-th ghost?
- $\frac{1}{K} \cdot (T_i - S_i) = X_i$
- $S_i = K \cdot X_i - T_i$

Ghost Encounters

Analysis

- The speed of the person is $\frac{1}{K}$.
- What is the condition of S that the person can encounter the i-th ghost?
- $\frac{1}{K} \cdot (T_i - S_i) = X_i$
- $S_i = K \cdot X_i - T_i$
- To maximize the total number of ghosts encountered, we need to pick the most frequent number among $K \cdot X_i - T_i$ $(1 \leq i \leq N)$.

Ghost Encounters

Analysis

- The speed of the person is $\frac{1}{K}$.
- What is the condition of S that the person can encounter the i-th ghost?
- $\frac{1}{K} \cdot (T_i - S_i) = X_i$
- $S_i = K \cdot X_i - T_i$
- To maximize the total number of ghosts encountered, we need to pick the most frequent number among $K \cdot X_i - T_i$ $(1 \leq i \leq N)$.
- Time complexity $O(N \cdot log(N))$ with a map.

Demo

Restructuring Company

There are N teams ($1 \leq N \leq 2e5$). Design a data structure that supports the following 3 types of queries ($1 \leq Q \leq 5e5$).

- Merge team X and team Y.
- Merge team in a range $[X, Y]$.
- Query if team X and team Y are merged together.

Restructuring Company

Naive approach

- Union-find.
- Type-1. Merge X and Y.
- Type-2. Merge X with $X+1$, $X+1$ with $X+2$, ..., $Y-1$ with Y.
- Type-3. Check if X and Y are in the same CC.

Restructuring Company

Naive approach

- Union-find.
- Type-1. Merge X and Y.
- Type-2. Merge X with $X+1$, $X+1$ with $X+2$, ..., $Y-1$ with Y.
- Type-3. Check if X and Y are in the same CC.
- Type-1 and 3 are $O(1)$ per query, type-2 is $O(N)$ per query.
- Time complexity: $O(N \cdot Q)$. Too slow!

Restructuring Company

Analysis

- Type-2 query is too slow.
- Merging $[X, Y]$ can be interpreted as merge X to all points in the range $[X+1, Y]$.

Restructuring Company

Analysis

- Type-2 query is too slow.
- Merging $[X, Y]$ can be interpreted as merge X to all points in the range $[X+1, Y]$.
- What data structure should we use for range operations?

Restructuring Company

Analysis

- Type-2 query is too slow.
- Merging $[X, Y]$ can be interpreted as merge X to all points in the range $[X+1, Y]$.
- What data structure should we use for range operations?
- Range tree!

Restructuring Company

Solution

- Recall that in a range tree, the root represents the range $[1, N]$.

Restructuring Company

Solution

- Recall that in a range tree, the root represents the range $[1, N]$.
- The parent represents the range $[l, r]$, the left-child contains the range $[l, \frac{l+r}{2}]$, the right-child contains the range $[\frac{l+r}{2} + 1, r]$.

Restructuring Company

Solution

- Recall that in a range tree, the root represents the range $[1, N]$.
- The parent represents the range $[l, r]$, the left-child contains the range $[l, \frac{l+r}{2}]$, the right-child contains the range $[\frac{l+r}{2} + 1, r]$.
- Merge $[X, Y]$ means merging X with all the "top-level" nodes representing the range $[X + 1, Y]$.

Restructuring Company

Solution (cont.)

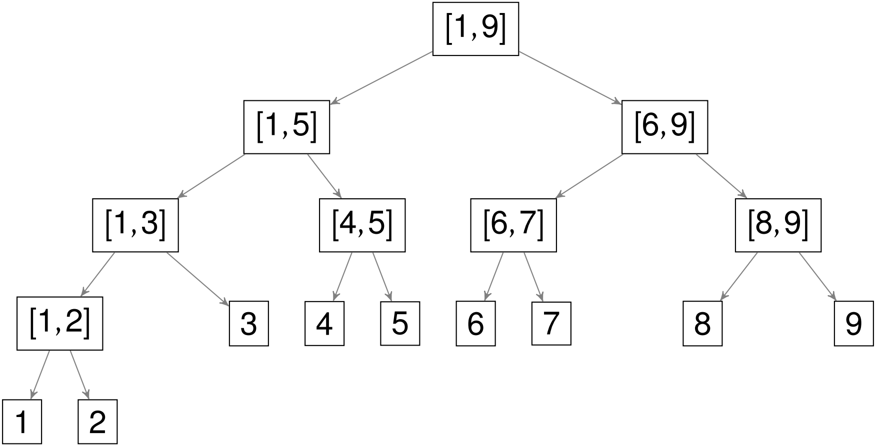
- Top-level node of a range $[s, e]$

void `query`(s, e, l, r, $index$):
    if $s \leq l \wedge r \leq e$ **then**
        $tree[index]$ is a top-level node
        return
    $mid = \frac{l+r}{2}$
    if $e \leq mid$ **then**
        $query(s, e, l, mid, index * 2)$
    else if $s \geq mid + 1$ **then**
        $query(s, e, mid + 1, r, index * 2 + 1)$
    else
        $query(s, e, l, mid, index * 2)$
        $query(s, e, mid + 1, r, index * 2 + 1)$

Restructuring Company

Example

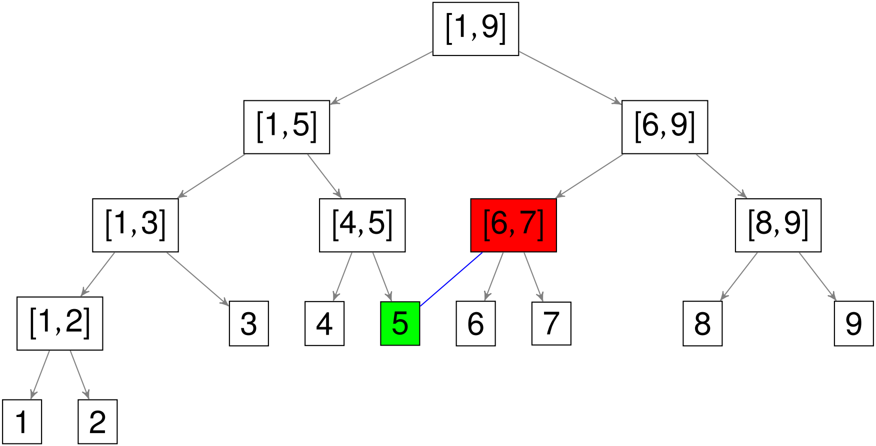
Merge range [5, 7] <==> Merge 5 with range [6, 7].



Restructuring Company

Example

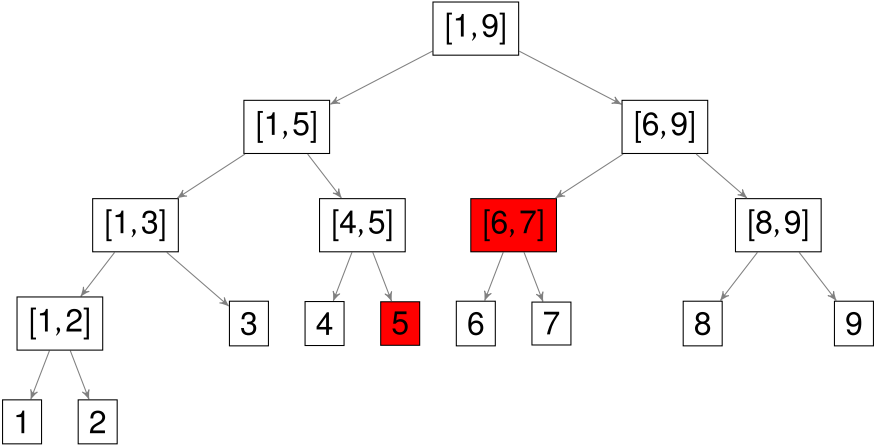
Merge range [5, 7] <==> Merge 5 with range [6, 7].



Restructuring Company

Example

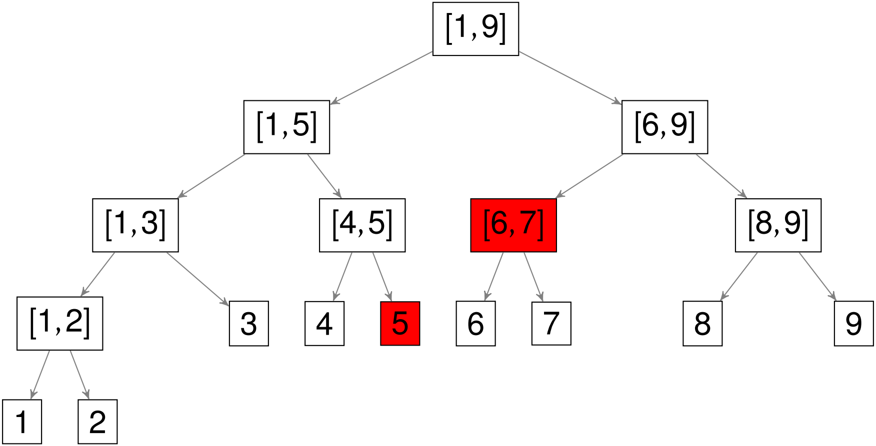
Is that all? No!



Restructuring Company

Example

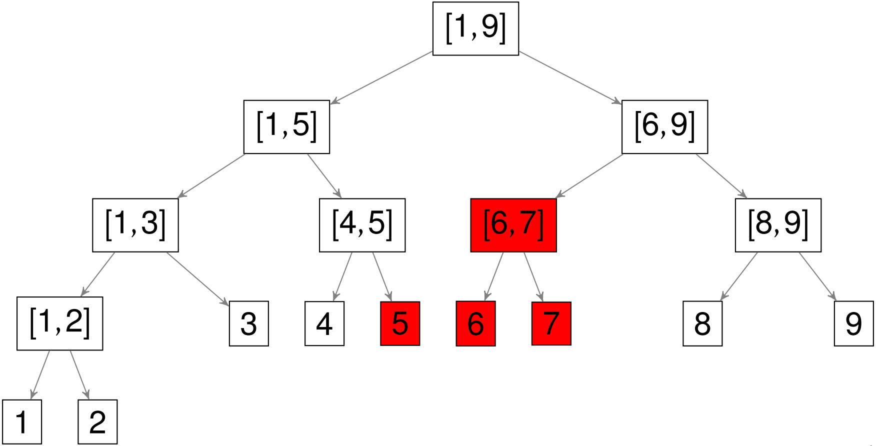
For example, 5 and 6 are not really merged.



Restructuring Company

Example

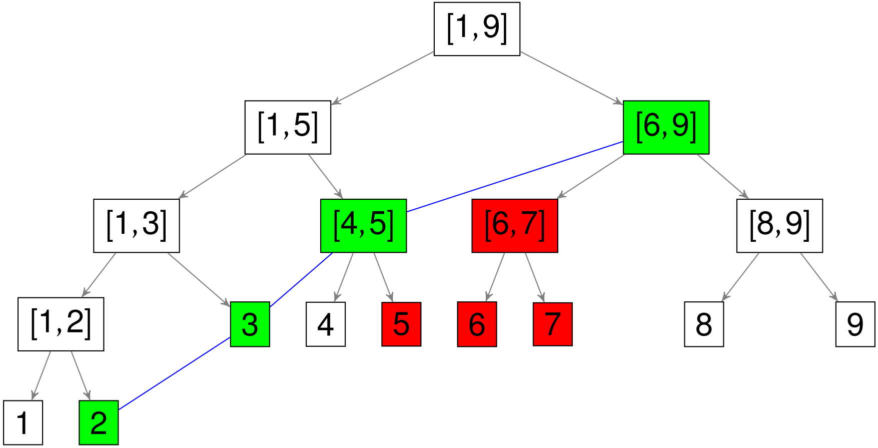
We need to propagate the range to the leaf, or until we meet a merged range.



Restructuring Company

Example

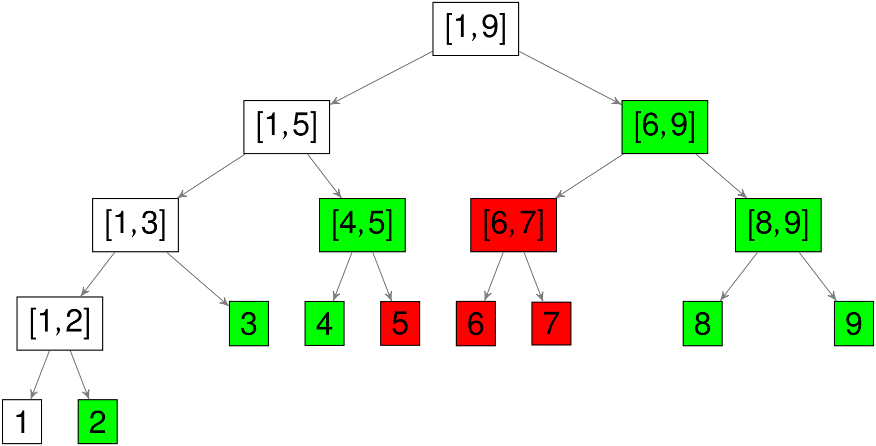
Merge 2 with [3, 9].



Restructuring Company

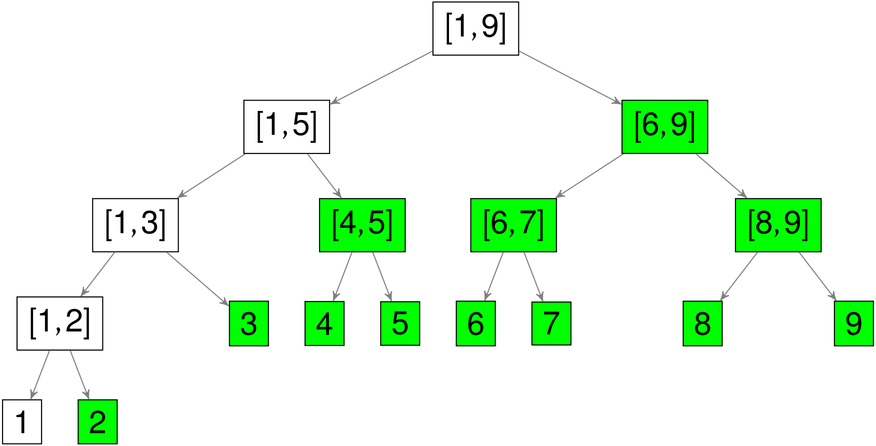
Example

Since [6, 7] are merged, don't need to merge all the way to the leaf!



Restructuring Company

Example



Demo