

## 128. Longest Consecutive Sequence

100, 7, 3, 6, 9, 8

### Union Find

- Stores a partition of a set into disjoint subsets
- Find a representative member of a set

Ⓐ Representative Array (Each rep is themselves before merging consecutive #)

[0, 1, 2, 3, 4, 5]

Ⓑ Map of element to index, for O(1) checking if  $n \pm 1$  are in the set.

100  $\rightarrow$  0  
7  $\rightarrow$  1  
3  $\rightarrow$  2  
6  $\rightarrow$  3  
9  $\rightarrow$  4  
8  $\rightarrow$  5

Ⓒ Initialize datastructures for union/find A, B

① Iterate through elements in array

check if neighbors  $n \pm 1$  are in it  $\rightarrow$  merge sets  
 $n - 1$  not in  $\rightarrow$  continue

### Union / Find Impl

find(x):

if parent[x] == x:

return x

parent[x] = find(parent[x])

return parent[x]

merge(x):

An optimization on this is for each node, we store metadata on how big each set is

An array all set to 1 suffices  $[1] \cdot \text{len}(A)$

```
parent_a, parent_b = find(a), find(b)
```

```
if parent_a == parent_b:
```

```
    return
```

```
rank_a, rank_b = rank[parent_a], rank[parent_b]
```

```
merged_rank = rank_a + rank_b
```

```
if rank_a >= rank_b:
```

```
    rank[parent_a] = merged_rank
```

```
    parent[parent_b] = parent_a
```

② Get the unique representative  $\forall x \in \text{nums}$

The one seen the most is LCS