

June 3<sup>rd</sup> 2025

problem: Maximum group fund for flood relief

Approaches:-

1. Initialize union-find structure:
  - parent[i] = i
  - rank[i] = 0

2. Union operations:

for each connection (U, V) union them

3. Group fund calculation

find root parent.

Add their fund to groupFunds[root].

4. Get Max fund.

pseudo code:-

int[] parent;

int[] rank;

unionfind (int size){

parent = new int [size];

rank = new int [size];

for (int i=0; i<size; i++) {  
parent[i] = i; }

int find (int u) { if (parent[u] != u) {

parent[u] = find (parent[u]); return parent[u]

}

void union (int u, int v) {

int rootU = find (u);

int rootV = find (v);

if (rootU == rootV) return;

if (rank[rootU] < rank[rootV]) parent[rootU] = rootV;

else { parent[rootU] = parent[rootV] = rootV;

rank[rootU]++;



Program parent = [0, 1, 2, 3, 4]

=>

• Union (0, 2) → volunteer 1 & 3

• " (1, 2) → " 2 & 3

• " (0, 1) → Already connected via 3

All 0 - 1 - 2 connected → parent: [0, 0, 0, 3, 4]

group 1: [0, 1, 2]

" 2: [3]

" 3: [4]

group led by parent 0 → funds =  $23 + 43 + 123 = 189$   
" " " " 3 → funds = 54  
" " " " 4 → fund = 2

Max = 189

## problem 2°

Approach: - take largest available fuel from stations you've passed only when you need to refuel

- when you move forward, push fuel of stations you pass into max-heap.
  - if you can't reach target: Refuel using starting with highest fuel from heap.
  - repeat until reach

```

pseudo code:
priorityQueue < integer > maxHeap = new
priorityQueue < > (Collections. reverseOrder ());
int stops = 0, i=0 ; n = station. length;
long fuel = startFuel;
while (fuel < target) {
    while (i < n && stations[i][0] <= fuel)
}

```

maxheap-offer(stations[i][i]); i++;

3

if (maneuver.isEmpty ()) return -1;

```
fuel += maxHeap.poll();  
stop++;
```

3

returns stops;

Very Run's      target = 100      start fuel = 10      stations = [(10, 60), (20, 30),  
(30, 30), (60, 40)]

1. At 0 with fuel = 10
  2. At 10, add 60 to heap  $\rightarrow$  fuel = 10
  3. Can't reach 20  $\rightarrow$  refuel with 60  $\rightarrow$  fuel 70
  4. Reach 20, add 30 to heap  $\rightarrow$  fuel = 70
  5. Reach 30, add 30 - fuel = 70
  6. Reach 60, add 40 - fuel = 70
  7. Can't reach 100  $\rightarrow$  refuel with 640  $\rightarrow$  fuel = 110  $\rightarrow$  reached stop = 2