COMP 202 FALL 2020 HOMEWORK 5

COMP-202 FALL 2020

Homework 5

I have completed this assignment individually, without support from anyone else. I hereby accept that only the below listed sources are approved to be used during this assignment:

(i) Course book

(ii) All material that is made available to me by professor (e.g. via Blackboard for this course website, email from professor/TA)

(iii) Notes taken by me during lecture

I have not used, accessed or taken any unpermitted information from any other source. Hence all effort belongs to me.

Sumeyra Erdogan

68912

```
playGame(num_players, portion, min_points, max_points, rounds, seed)
  rand = new Random(seed)
  ArrayPriorityQueue maxHeap = new ArrayPriorityQueue(num_players,false)
  ArrayPriorityQueue minHeap =  new ArrayPriorityQueue(num_players,true)

  for(i to num_players)


    randomInt = rand.nextInt(max_points - min_points) + min_points;
    player = Player(i, randomInt)


    player.setMaxIndex(i+1)
    player.setMinIndex(i+1)


    minHeap.add(player)
    maxHeap.add(player)

  for(i to rounds)


    minPlayer = minHeap.poll();
    maxPlayer = maxHeap.poll();


    points_transferred = (maxPlayer.points * portion);


    minPlayer.points += points_transferred
    maxPlayer.points -= points_transferred


    minHeap.removeAtIndex(maxPlayer.minIndex)
    maxHeap.removeAtIndex(minPlayer.maxIndex)


    minHeap.add(maxPlayer)
    minHeap.add(minPlayer)


    maxHeap.add(maxPlayer)
    maxHeap.add(minPlayer)



  printPlayersById(maxHeap.getHeap())
```

Complexity for playGame function:

We allocate memory for two array-based heaps with size num_players. It takes O(1) time and O(num_players) space.

Num_players times we randomly assign a int to a player and add it to the heap. Add() method of heap takes O(logn) times. We do this operation num_players times so, it takes num_players * logn time which is O(n).

Every round the algorith do the following steps;

-Poll the max and min players from heaps.

Poll operation takes O(logn) time.

-Calculates the point transferred.

Calculation takes O(1) time.

-Adds the point transferred to min player ands subtract it from max player.

Adding and subtracting takes O(1) times.

-Removes max player from minHeap and remove min player from maxHeap using the minIndex and maxIndex information of Players.

Removing an element from specified heap index takes O(logn) time.

-Adds new min player and new max player to min and max heaps.

Adding to a heap takes O(logn) time.

So every round takes O(2logn) + O(1) + O(1) + O(2logn) + O(4logn) which is just O(logn).

For printing we iterate over the heap array, so it is O(n).

See the ArrayPrioirtyClass.java to see the detailed heap implementation.