Programming Assignment HW4

The Professional Golf Association (PGA) is keen on tracking analytics and they've hired you to compute historical score statistics. The PGA has a list of per-round scores, indexed by time. For a given time interval *start..end*, they would like to know the golfer with the lowest scoring round during this interval. More formally, Given a list $S[1..n] = [(player, score)_i]_{i=1..n}$ of player-score tuples, the PGA wants to efficiently answer the query $\textsc{LowestPlayer}(start, end)$: "who was the player with the lowest score in $S[start..end]$?"

Part 1

You will create a data structure to compute $\textsc{LowestPlayer}(\cdot)$queries in $O(\log n)$ time. The data structure should take $O(n)$ time to compute and use $O(n)$ space.

To achieve this, your data structure will store the lowest scoring player and their score for time intervals of the form $\left[x \cdot 2^i, \min\left(n, (x+1) \cdot 2^i\right)\right]$ where $i = 0..\lceil \log(n) \rceil$,$x = 1..\left(\frac{n}{2^i} - 1\right)$.

Suppose that $T[x,i]$ holds the tuple (p,s) where p is the lowest scoring player, and s is their score in $S\left[x \cdot 2^i..(x+1) \cdot 2^i\right]$.

1. Given an interval *start..end*, explain how T can be used to compute the lowest scoring player in $O(\log n)$ time (2 points).
2. Show how to compute $T[x,0]$ (1 point).
3. Give a (constant time) recurrence for computing $T[x,i]$. State any base cases or boundary conditions (1 points).
4. Give the order that the table should be filled in (1 point).
5. Complete the code for part 1 by filling in the "TODO's" (5 points)

Part 2

The PGA is flush with cash, and they're willing to buy you more memory in order to speed up the time complexity of $\textsc{LowestPlayer}(\cdot)$ queries. In this part, you will devise a data structure that uses more space, but can answer $\textsc{LowestPlayer}(\cdot)$ queries in constant time.

Your new data structure will use two tables to store the data for time intervals of the form (i) $\left[x, \min\left(n, x+2^i\right)\right]$, and (ii)$\left[\max\left(1, x-2^i\right), x\right]$ where $x = 1..n$ $i = 0..\lceil \log(n-x) \rceil$ for (i) and $i = 0..\lceil \log(x) \rceil$ for (ii).

That is,

$$T_1[x,i] = \min_s \left\{(s,p) \in S\left[x..\min\left(n, x+2^i\right)\right]\right\}$$

and

$$T_2[x,i] = \min_s \left\{(s,p) \in S\left[\max\left(1, x-2^i\right), x\right]\right\}$$

1. Analyze the space complexity of this new data structure (1 point)
2. Describe how to compute the entries for the case of $i = 0$ (1 point)
3. Describe how to use $T_1$and T_2 to compute LowestPlayer($\cdot$) queries in constant time (2 points)
4. Give a (constant time) recurrence for computing $T[x, i]$, and state any boundary conditions and/or base cases (1 point)
5. Complete the code for part 2 by filling in the "TODO's" (5 points)

General

You may include your answers to the questions (1-4) for each part in the space provided in the code.