Home Work № 5

CSCI B505 Applied Algorithms 10/02/2022

Instructions

1. Under any circumstances the deadline will NOT be extended.
2. Code should be written in python language in a Jupyter notebook .
3. Unless mentioned in the questions, feel free to use built-in python functions
4. Upload Jupyter notebook and html file of the Jupyter notebook in canvas. No other forms submissions is considered as valid submission. Make sure to have the output of the required cells of the jupyter notebook and its html version before making submission.
5. Plagiarism is unacceptable and we have ways to find it. So do not do it.
6. For Problems 2, 3 and 4 , use pen and paper and upload the same as pdf.
7. There are five problems. Problem 1 - 30 points, problems 2 and 4 - 15 points each, problem 3 and problem 5 - 20 points each.
8. The code should readable with variables named meaningfully
9. Write test cases wherever required so that they cover all scenarios.

Problem 1

Let us implement the amoritized dictionary in python as a class using python dicitonary. The dictionary keys acts levels and values of the dicitonaries are list of elements. The level i contains either 2^i elements or no elements at all. If the level has elements , then all the elements are in sorted order.For this class implement search, insert, print methods.

Example :

```
ad_obj = amor_dict([23, 12 ,24, 42])

ad_obj.print()
0:empty
1:empty
2:[12, 23, 24, 42]

ad_obj.insert(11)
```

```
ad_obj.print()
0:[11]
1:empty
2:[12, 23, 24, 42]

ad_obj.insert(74)
ad_obj.print()
0:empty
1:[11, 74]
2:[12, 23, 24,42]

ad_obj.search(74)
level 1

ad_obj.search(77)
level -1
```

```
class amor_dict():
      def __init__(self):
          ## your code here
          pass
      def insert(self, num):
          ## code here
          pass
      def search(self, num):
          ## code here
          pass
      def print(self):
          ## code here
          pass
```

Problem 2

Prove the following for the amortized dictionaries if it has n elements

- The insert method will have amortized cost of $O(logn)$
- The search method will have time complexity of $O(log^2 n)$

Problem 3

Using master theorem and with justification, find the time complexity of the following recurrence relations.

1. $T(n) = 8 * T(n/3) + 2^n$
2. $T(n) = 3 * T(n/3) + n/2$
3. $T(n) = 2 * T(n/4) + \sqrt{n}$
4. $T(n) = 4 * T(n/2) + 3 * n$

Problem 4

When a sequence of n operations are performed on a special data structure the operation costs i , if i is a perfect square or else zero. Using aggregate analysis, find the amortized cost per operation in terms of n.

Problem 5

Lydia being a professor would like to explain students a concept of sequence of digit strings, given integer n, return the nth term of the count digit sequence, Let's help students in implementing a recursion approach for the given problem. For example, the given digit string "3322251" can be interpreted as two 3's, three 2's, one 5 and one 1, In numerical string form this can be written "23321511", Now for any given integer n, return the nth term of $count\_digit\_string$ sequence.

```
Example 1:

Input n = 1
Output: ‘‘1’’
This is the Base case

Example 2:

Input n= 4
Output: ‘‘1211’’

Explanation:

count_digit_string(1) = ‘‘1’’

count_digit_string(2) = Number of ‘‘1’’ = one 1 = ‘‘11’’
```

count_digit_string(3) = Number of ‘‘11’’ = two 1’s = 21

count_digit_string(4) = Number of ‘‘21’’ = one 2 and one 1 = ‘‘1211’’

Constraints: 1<= n<= 30

```
def count_digit_string(n):
    #
    #####logic####
    #
   return nth term count_digit_string
```