CS/ECE 374 P16

Junquan Chen, Jiawei Tang, Pengxu Zheng

TOTAL POINTS

7 / 100

QUESTION 1

1 Problem 16.A. 7 / 70

✓ **+ 7 pts Correct English Description of quantity being computed. If not present, the rest of the answer is invalid.**

+ 28 pts Combined Rubric Item: Correct Base Case (+7.0) and Correct Recursion (+21.0)

+ 7 pts Correct Base Case for the quantity.

+ 4 pts Minor Mistake in Base Case for the quantity

+ 21 pts Correct Recursive Definition

+ 14 pts One Minor Bug in Recursive Definition

+ 7 pts Recursive Definition in right track, but has multiple errors

+ 28 pts Combined Rubric Item: Correct Data Structure (+7.0), Evaluation Order (+14.0), Final Computation (+7.0)

+ 7 pts Correct Data Structure for quantity is stated

+ 14 pts Correct Evaluation Order stated

+ 7 pts Correctly stated how to compute final answer

+ 7 pts Correct Runtime Analysis

+ 17.5 pts IDK

- 15 pts No clear English description of function, or incorrect base cases and recurrence

- 10 pts Using code (that is hard to read) rather than pseudocode

+ 0 pts Incorrect; Not understanding the question (see comments below)

Which is recursive function calling itself with the same parameters? How can sum[i][j] be initialized in the beginning if f(xj) is still unknown?

QUESTION 2

2 Problem 16.B. 0 / 30

+ 7.5 pts IDK

+ 10 pts Correct Auxillary quantity which needs to be computed. If this is absent, the rest of the answer gets 0 points.

+ 17 pts Correct description of Backtracking idea

+ 3 pts Correct Runtime stated

✓ **+ 0 pts Totally incorrect (see comments below) or blank**

HW6 Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- ≪**Pengxu Zheng**≫: ≪**pzheng5**≫
- ≪**Junquan Chen**≫: ≪**junquan2**≫
- ≪**Jiawei Tang**≫: ≪**jiaweit2**≫

16

Solution:

16.A.

We define the function MTE to compute the minimal total error of the step function with i bounded in [0, k - 1] and j bounded in [1, n], where k and n refer to the same variables in the problem statement. The following recurrence demonstrates the MTE function:

$$MTE(i,j) = \begin{cases} 0, & j = 1 \\ min(MTE(i+1, j+1) + |y_j - f(x_i)|), MTE(i,j), & otherwise \end{cases}$$

The pseudocode is as follows:

```
initialize arrays minErr[n], sum[k][n];
for i = 0 to k - 1:
    for j = 1 to n:
        sum[i][j] += abs(f(x_j) − y_j); // f(x) here denotes the value of the step function at x

for i = 0 to k - 1:
    for j = 1 to n:
        minErr += min(sum[i][j]);

int counter = 0;
for all x in minErr:
    counter += minErr[x];
return counter;
```

Since the dominant factor of runtime is bounded by k*(n - 1) accesses to sum[i][j], we conclude that the algorithm is bounded by O(kn) time.

16.B.

To calculate the step function itself, we can use a sorted array to hold the y coordinates of the data points. After that, we apply MTE funciton to each of the data points. We divide the codomain y to k - 1 intervals such that each interval is determined by largest value returned by MTE function. The maximum y value in each interval will then be selected to be the step function.

1 Problem 16.A. 7 / 70

✓ **+ 7 pts Correct English Description of quantity being computed. If not present, the rest of the answer is invalid.**

+ **28 pts** Combined Rubric Item: Correct Base Case (+7.0) and Correct Recursion (+21.0)
+ **7 pts** Correct Base Case for the quantity.
+ **4 pts** Minor Mistake in Base Case for the quantity
+ **21 pts** Correct Recursive Definition
+ **14 pts** One Minor Bug in Recursive Definition
+ **7 pts** Recursive Definition in right track, but has multiple errors
+ **28 pts** Combined Rubric Item: Correct Data Structure (+7.0), Evaluation Order (+14.0), Final Computation (+7.0)
+ **7 pts** Correct Data Structure for quantity is stated
+ **14 pts** Correct Evaluation Order stated
+ **7 pts** Correctly stated how to compute final answer
+ **7 pts** Correct Runtime Analysis
+ **17.5 pts** IDK
- **15 pts** No clear English description of function, or incorrect base cases and recurrence
- **10 pts** Using code (that is hard to read) rather than pseudocode
+ **0 pts** Incorrect; Not understanding the question (see comments below)

Which is recursive function calling itself with the same parameters? How can sum[i][j] be initialized in the beginning if f(xj) is still unknown?

HW6 Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- ≪**Pengxu Zheng**≫: ≪**pzheng5**≫
- ≪**Junquan Chen**≫: ≪**junquan2**≫
- ≪**Jiawei Tang**≫: ≪**jiaweit2**≫

16

Solution:

16.A.

We define the function MTE to compute the minimal total error of the step function with i bounded in [0, k - 1] and j bounded in [1, n], where k and n refer to the same variables in the problem statement. The following recurrence demonstrates the MTE function:

$$MTE(i,j) = \begin{cases} 0, & j = 1 \\ min(MTE(i+1, j+1) + |y_j - f(x_i)|), MTE(i,j), & otherwise \end{cases}$$

The pseudocode is as follows:

```
initialize arrays minErr[n], sum[k][n];
for i = 0 to k - 1:
    for j = 1 to n:
        sum[i][j] += abs(f(x_j) − y_j); // f(x) here denotes the value of the step function at x

for i = 0 to k - 1:
    for j = 1 to n:
        minErr += min(sum[i][j]);

int counter = 0;
for all x in minErr:
    counter += minErr[x];
return counter;
```

Since the dominant factor of runtime is bounded by k*(n - 1) accesses to sum[i][j], we conclude that the algorithm is bounded by O(kn) time.

16.B.

To calculate the step function itself, we can use a sorted array to hold the y coordinates of the data points. After that, we apply MTE funciton to each of the data points. We divide the codomain y to k - 1 intervals such that each interval is determined by largest value returned by MTE function. The maximum y value in each interval will then be selected to be the step function.

2 Problem 16.B. 0 / 30

+ **7.5 pts** IDK

+ **10 pts** Correct Auxillary quantity which needs to be computed. If this is absent, the rest of the answer gets 0 points.

+ **17 pts** Correct description of Backtracking idea

+ **3 pts** Correct Runtime stated

✓ **+ 0 pts** **Totally incorrect (see comments below) or blank**