CS/ECE 374 P15

Pengxu Zheng, Junquan Chen, Jiawei Tang

TOTAL POINTS

95 / 100

QUESTION 1

1 Problem 15.A. **10 / 10**

✓ **- 0 pts** Correct

QUESTION 2

2 Problem 15.B. **20 / 20**

✓ **- 0 pts** Correct

QUESTION 3

3 Problem 15.C. **15 / 20**

✓ **- 5 pts** **Algorithm runs slower than O(n log n)**

QUESTION 4

4 Problem 15.D. **50 / 50**

✓ **- 0 pts** Correct

HW Solution

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

Submitted by:

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

15

Solution:

15.A. Since all weights are equal, the minimized weighted distance can be written as

$$d = \sum_{i}^{n} w|x_j - x_i|$$

$$d = w\sum_{i}^{n} |x_j - x_i| = w \cdot t$$

We want to prove that when x_j is the median, we have the minimized $t = \sum_i^n |x_j - x_i|$ so that we can have the minimized d. There are two cases to consider. We assume that m is the median of X.
First, assume there is a x_k, where $x_j > x_k \geq m$ and x_k is the next number that is smaller than x_j. We have $t = \sum_i^n |x_j - x_i|$ and $t_2 = \sum_i^n |x_k - x_i|$. Because

$$t = t_2 + |x_j - x_k|(k - (n - j))$$

and $(k - (n - j)) = k - n + j = k + j - n > 0$, we can prove that $t > t_2$. Therefore, x_j must not be larger than m.
Second, assume there is a x_l, where $x_j < x_l \leq m$ and x_l is the next number that is larger than x_j. Use the same approach as in the first case, we can prove that x_j must not be smaller than m.
Thus, x_j must be equal to m.

15.B. At the beginning we sort X such that for every index i, k, if $i < k$, then $x_i < x_k$. We have x_j to be the median of X.
First we want to prove $\sum_{x_i > x_j} w_i = \sum_{i>j} w_i \leq \frac{1}{2}$. $j + 1$ is the next number that is larger than j. Then we have

$$\sum_i w_i|x_{j+1} - x_i| = \sum_i w_i|x_j - x_i| + \sum_{i\leq j} w_i(x_j - x_{j+1}) - \sum_{i>j} w_i(x_j - x_{j+1})$$

Since $\sum_i w_i|x_{j+1} - x_i| > \sum_i w_i|x_j - x_i|$ by definition of x_j, therefore $\sum_{i\leq j} w_i(x_j - x_{j+1}) > \sum_{i>j} w_i(x_j - x_{j+1})$. We then can say that $\sum_{i\leq j} w_i > \sum_{i>j} w_i$. Also, we know that $\sum_{i\leq j} w_i + \sum_{i>j} w_i = 1$. Therefore, $2\sum_{i>j} w_i < 1$. $\sum_{i>j} w_i < \frac{1}{2}$. Thus, $\sum_{i>j} w_i \leq \frac{1}{2}$ is true.
Second, we want to prove $\sum_{x_i < x_j} w_i = \sum_{i<j} w_i < \frac{1}{2}$. $j - 1$ is the next number that is smaller than j. Then we have

$$\sum_i w_i|x_{j-1} - x_i| = \sum_i w_i|x_j - x_i| - \sum_{i<j} w_i(x_j - x_{j-1}) + \sum_{i\geq j} w_i(x_j - x_{j-1})$$

Since $\sum_i w_i|x_{j-1} - x_i| > \sum_i w_i|x_j - x_i|$ by definition of x_j, therefore $\sum_{i<j} w_i(x_j - x_{j-1}) < \sum_{i\geq j} w_i(x_j - x_{j-1})$. We then can say that $\sum_{i<j} w_i < \sum_{i\geq j} w_i$. Also, we know that $\sum_{i<j} w_i + \sum_{i\geq j} w_i = 1$. Therefore, $2\sum_{i<j} w_i < 1$. Thus, $\sum_{i<j} w_i < \frac{1}{2}$ is true.

1 Problem 15.A. **10 / 10**

✓ **- 0 pts** Correct

HW Solution

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

Submitted by:

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

15

Solution:

15.A. Since all weights are equal, the minimized weighted distance can be written as

$$d = \sum_{i}^{n} w|x_j - x_i|$$

$$d = w\sum_{i}^{n} |x_j - x_i| = w \cdot t$$

We want to prove that when x_j is the median, we have the minimized $t = \sum_i^n |x_j - x_i|$ so that we can have the minimized d. There are two cases to consider. We assume that m is the median of X.
First, assume there is a x_k, where $x_j > x_k \geq m$ and x_k is the next number that is smaller than x_j. We have $t = \sum_i^n |x_j - x_i|$ and $t_2 = \sum_i^n |x_k - x_i|$. Because

$$t = t_2 + |x_j - x_k|(k - (n - j))$$

and $(k - (n - j)) = k - n + j = k + j - n > 0$, we can prove that $t > t_2$. Therefore, x_j must not be larger than m.
Second, assume there is a x_l, where $x_j < x_l \leq m$ and x_l is the next number that is larger than x_j. Use the same approach as in the first case, we can prove that x_j must not be smaller than m.
Thus, x_j must be equal to m.

15.B. At the beginning we sort X such that for every index i, k, if $i < k$, then $x_i < x_k$. We have x_j to be the median of X.
First we want to prove $\sum_{x_i > x_j} w_i = \sum_{i>j} w_i \leq \frac{1}{2}$. $j+1$ is the next number that is larger than j. Then we have

$$\sum_i w_i|x_{j+1} - x_i| = \sum_i w_i|x_j - x_i| + \sum_{i \leq j} w_i(x_j - x_{j+1}) - \sum_{i>j} w_i(x_j - x_{j+1})$$

Since $\sum_i w_i|x_{j+1} - x_i| > \sum_i w_i|x_j - x_i|$ by definition of x_j, therefore $\sum_{i\leq j} w_i(x_j - x_{j+1}) > \sum_{i>j} w_i(x_j - x_{j+1})$. We then can say that $\sum_{i\leq j} w_i > \sum_{i>j} w_i$. Also, we know that $\sum_{i\leq j} w_i + \sum_{i>j} w_i = 1$. Therefore, $2\sum_{i>j} w_i < 1$. $\sum_{i>j} w_i < \frac{1}{2}$. Thus, $\sum_{i>j} w_i \leq \frac{1}{2}$ is true.
Second, we want to prove $\sum_{x_i < x_j} w_i = \sum_{i<j} w_i < \frac{1}{2}$. $j - 1$ is the next number that is smaller than j. Then we have

$$\sum_i w_i|x_{j-1} - x_i| = \sum_i w_i|x_j - x_i| - \sum_{i<j} w_i(x_j - x_{j-1}) + \sum_{i\geq j} w_i(x_j - x_{j-1})$$

Since $\sum_i w_i|x_{j-1} - x_i| > \sum_i w_i|x_j - x_i|$ by definition of x_j, therefore $\sum_{i<j} w_i(x_j - x_{j-1}) < \sum_{i\geq j} w_i(x_j - x_{j-1})$. We then can say that $\sum_{i<j} w_i < \sum_{i\geq j} w_i$. Also, we know that $\sum_{i<j} w_i + \sum_{i\geq j} w_i = 1$. Therefore, $2\sum_{i<j} w_i < 1$. Thus, $\sum_{i<j} w_i < \frac{1}{2}$ is true.

2 Problem 15.B. 20 / 20

✓ **- 0 pts** Correct

15.C. We will have two loops. The outer loop is to set a x to x_j per iteration. The inner loop is to calculate $\sum_i^n w_i|x_j - x_i|$ for every x_j we choose in the outer loop. Then we can have every possible result of weighted distance. Thus we can find the min which is the minimized weighted distance. We therefore can find the corresponding x_j. The running time will be $O(n^2)$ because we iterate through every possible x_j and in each iteration, we will iterate through every elements in the array to calculate weighted distance. In short, we have outer loop iterating through the whole array and the inner loop also iterating through the whole array.

15.D. Here we are trying to find weighted median. Basically we want to find a x_j that satisfies the conditions in 15.B. For this part, we are using an algorithm described in class. The algorithm can find the median in $O(n)$ time. Here we are defining such algorithm to be findMedian(Array).

```
findWeightedMedian(X,W){
    if(X.length<3){
        return X[i] with the minimum weight
    }
    x_j = findMedian(X)
    s1 = Σ_{X[i]<x_j} w[i]
    s2 = Σ_{X[i]>x_j} w[i]
    if (s1>=1/2){
        return findWeightedMedian(left half of X, left half of W) //recursion
    }
    else if(s2>1/2){
        return findWeightedMedian(right half of X, right half of W) //recursion
    }
    else{//satisfy conditions in 15.B.
        return x_j
    }
}
```

We have the base case as $T(n) = O(1)$ when $n < 3$. Otherwise, we have $T(n) = T(n/2)+cn$. By unrolling, we can have the result that $T(n) = T(n/4) + cn + cn/2 \; ... \; = T(\frac{n}{2^i}) + cn \cdot 2 = O(n)$.

3 Problem 15.C. 15 / 20

✓ **- 5 pts Algorithm runs slower than O(n log n)**

15.C. We will have two loops. The outer loop is to set a x to x_j per iteration. The inner loop is to calculate $\sum_i^n w_i|x_j - x_i|$ for every x_j we choose in the outer loop. Then we can have every possible result of weighted distance. Thus we can find the min which is the minimized weighted distance. We therefore can find the corresponding x_j. The running time will be $O(n^2)$ because we iterate through every possible x_j and in each iteration, we will iterate through every elements in the array to calculate weighted distance. In short, we have outer loop iterating through the whole array and the inner loop also iterating through the whole array.

15.D. Here we are trying to find weighted median. Basically we want to find a x_j that satisfies the conditions in 15.B. For this part, we are using an algorithm described in class. The algorithm can find the median in $O(n)$ time. Here we are defining such algorithm to be findMedian(Array).

```
findWeightedMedian(X,W){
    if(X.length<3){
        return X[i] with the minimum weight
    }
    x_j = findMedian(X)
    s1 = Σ_{X[i]<x_j} w[i]
    s2 = Σ_{X[i]>x_j} w[i]
    if (s1>=1/2){
        return findWeightedMedian(left half of X, left half of W) //recursion
    }
    else if(s2>1/2){
        return findWeightedMedian(right half of X, right half of W) //recursion
    }
    else{//satisfy conditions in 15.B.
        return x_j
    }
}
```

We have the base case as $T(n) = O(1)$ when $n < 3$. Otherwise, we have $T(n) = T(n/2) + cn$. By unrolling, we can have the result that $T(n) = T(n/4) + cn + cn/2 \; ... \; = T(\frac{n}{2^i}) + cn \cdot 2 = O(n)$.

4 Problem 15.D. **50 / 50**

✓ **- 0 pts** Correct