Homework 2

Ege Aktemur

November 2022

Question 1

Question 1.a Sort and Pick

Question 1.a.1 Sort

Comparison-based algorithms are algorithms that sorts the given array in specific order by only using comparison. Even though they might have best case lower bound can be $\omega(n)$ like Insertion Sort, They generally have running time $\theta(n * \log n)$ as an average case. One of these algorithms is Merge Sort which has a worst case running time of $\theta(n * \log n)$. Which can be proven with Master Theorem as we saw in the lecture. Since formula of the merge sort is $T(n) = 2 * T(n/2) + \theta(n)$ we can use Master Theorem because a = 2, b = 2 and f(n) = $\theta(n)$ which is asymptotically positive. Therefore $(n^{\log_b a}) = (n^{\log_2 2}) = n$. Using this we found that we should use Case 2 which concludes that Merge Sort has a worst case running time of $\theta(n * \log n)$.

Question 1.a.2 Pick

After we sorted the list we will just get the first k elements which has a complexity of $\theta(k)$.

Question 1.a.3 Conclusion

Using the information prepared total complexity is $\theta(n * \log n) + \theta(k) = \theta(n * \log n + k)$.

Question 1.b Pick and Sort

Question 1.b.1 Pick

As we deeply analysed in the lecture and quizzes, groups of 5 in order statistic method works in $\theta(n)$ time which is better than groups of 3 and 7. Because of this first part has $\theta(n)$ complexity.

Question 1.b.2 Sort

In the second part we will sort k number of elements. Again using Merge Sort as sorting algorithm, this step would cost $\theta(k * \log k)$.

Question 1.a.3 Conclusion

Using the information prepared total complexity is $\theta(n) + \theta(k * \log k) = \theta(n + k * \log k)$.

Question 1.c Conclusion

To sum up first method costs $\theta(n * \log n + k)$ and second one costs $\theta(n + k * \log k)$. Since we can not get more numbers than the length of the array, k must be smaller or equal to n. Because of this., second method will be always work better or same to the first method. This is why I would use the second method.

Question 2

Question 2a Modifying Radix Sort

Question 2a1 Modifications in Radix Sort

Similar to radix sort for integers we should start from the rightmost character and go to leftmost. In the radix sort for integers we must insert zero for the leftmost characters to make the algorithm work for different number digits. However, for the strings we must add characters to the rightmost and make lengths of the words same. I will add ”_” characters in the solution.

Question 2a2 Modifications in Count Sort

There are not many modifications I would make in Count Sort. I would fix the size of the storage of Count Sort as 27 which is ”_” character plus the alphabet and the number in C will correspond to [”_”,”A”,”B” ,... ,”Y”,”Z”].

Question 2b Illustration of Sorting

Step 1: We want to sort the array A=[”Y”,_”,”_”,”_”,”_”].
We will set each score of the Count Sort array as 0.
Then increase the amount of ”Y” by 1.
Then increase the amount of ”_” by 1.
Then increase the amount of ”_” by 1.
Then increase the amount of ”_” by 1.
Then increase the amount of ”_” by 1.
Which will make the resulting Count Sort Array as:
Format of C = [”_”,”A”,”B” ,... ,”Y”,”Z”]
C = [4 , 0 , 0 , ... , 1 , 0]
C’ = [4 , 4 , 4 , ... , 5 , 5]
Then we will insert to B accordingly.
B = [, , , _,] decrease ”_”’s value by 1 Therefore, C’ will become [3, ... 5, 5]
B = [, , _, _,] decrease ”_”’s value by 1 Therefore, C’ will become [2, ...5, 5]
B = [, _, _, _,] decrease ”_”’s value by 1 Therefore, C’ will become [1, ... 5, 5]
B = [_, _, _, _,] decrease ”_”’s value by 1 Therefore, C’ will become [0, ... 5, 5]
B = [_, _, _, _, Y] decrease ”Y”’s value by 1 Therefore, C’ will become [0, ... 4, 5]
Then it result will be [”_”,”_”,”_”,”_”,”Y”]. So main array will be [”GORKEM _”,”GIRAY _ _”,”TAHIR _ _,”BARIS _ _”,”BATURAY”]

Step 2: We want to sort the array A=[”M”,_”,”_”,”_”,”A”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [3 , 1 , ... , 1 , ... , 0].
Which makes C’= [3 , 4 ... 5 .... 5].
Then we must insert characters to B accordingly. Which makes B [”_”,”_”,”_”,”A”,”M”]. So main array will be [”GIRAY _ _”,”TAHIR _ _,”BARIS _ _”,”BATURAY”,”GORKEM _”]

Step 3: We want to sort the array A=[”Y”,R”,”S”,”R”,”E”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [0 , ... , 1 , ... , 2 , 1 , ... ,0].
Which makes C’= [0 , ... , 1 , ... , 3 , 4 , ... ,5].
Then we must insert characters to B accordingly. Which makes B [”E”,”R”,”R”,”S”,”Y”]. So main array will be [”GORKEM _”, ”TAHIR _ _”,”BATURAY”,”BARIS _ _”, ”GIRAY _ _”]

Step 4: We want to sort the array A=[”K”,I”,”U”,”I”,”A”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [0 , 1 , ... , 2 , ... , 1 , ... ,1, ... , 0].
Which makes C’= [0 , 1 , ... , 3 , ... , 4 , ... ,5, ... , 5].
Then we must insert characters to B accordingly. Which makes B [”A”, ”I”, ”I”, ”K”, ”U”].
So main array will be [”GIRAY _ _”,”TAHIR _ _,”BARIS _ _”,”GORKEM _”,”BATURAY”]

Step 5: We want to sort the array A=[”R”, ”H”, ”R”, ”R”, ”T”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [0 , ... , 1 , ... , 3 , 1 , ... , 0].
Which makes C’= [0 , ... , 1 , ... , 4 , 5, ... , 5].
Then we must insert characters to B accordingly. Which makes B [”H”, ”R”,”R”, ”R”, ”T”].
So main array will be [”TAHIR _ _,”GIRAY _ _”,”BARIS _ _”,”GORKEM _”,”BATURAY”]

Step 6: We want to sort the array A=[”A”, ”I”, ”A”, ”O”, ”A”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [0 , 3 , ... , 1 , ... , 1 , ... , 0].
Which makes C’= [0 , 3 , ... , 4 , ... , 5 , ... , 5].
Then we must insert characters to B accordingly. Which makes B [”A”, ”A”,”A”, ”I”, ”O”].
So main array will be [”TAHIR _ _,”BARIS _ _”,”BATURAY”,”GIRAY _ _”,”GORKEM _”]

Step 7: We want to sort the array A=[”T”, ”B”, ”B”, ”G”, ”G”].
We will set each score of the Count Sort array as 0. After that, we will increase the amount of characters in the C array accordingly.
C array will become [0 , 0 , 2 ... , 2 , ... , 1 , ... , 0].
Which makes C’= [0 , 0 , 2 ... , 4 , ... , 5 , ... , 5].
Then we must insert characters to B accordingly. Which makes B [”B”, ”B”,”G”, ”G”, ”T”].
So main array will be [”BARIS _ _”,”BATURAY”,”GIRAY _ _”,”GORKEM _”,”TAHIR _ _”]

To explain more visually I prepared a table which shows which characters will be sorted **next** with the red characters.

Initial	Step=0	Step=1	Step=2	Step=3	Step=4	Step=5	Step=6	Step=7 (End)
BATURAY	BATURAY	GORKEM _	GIRAY _ _	GORKEM_	GIRAY _ _	TAHIR _ _	TAHIR _ _	BARIS _ _
GORKEM_	GORKEM_	GIRAY _ _	TAHIR _ _	TAHIR _ _	TAHIR _ _	GIRAY _ _	BARIS _ _	BATURAY
GIRAY _ _	GIRAY _ _	TAHIR _ _	BARIS _ _	BATURAY	BARIS _ _	BARIS _ _	BATURAY	GIRAY _ _
TAHIR _ _	TAHIR _ _	BARIS _ _	BATURAY	BARIS _ _	GORKEM_	GORKEM_	GIRAY _ _	GORKEM_
BARIS _ _	BARIS _ _	BATURAY	GORKEM_	GIRAY _ _	BATURAY	BATURAY	GORKEM_	TAHIR _ _

Question 2c Analysis of Sorting

There are three factors that affects the complexity of the sort. To make the analysis simpler I will define them first then explain why and how they effect.

1. As it is clear amount of words does effect the running time which will be showed with m.
2. Then for each word we will look for each character which is equal to the length of the longest word because of the changes I made in Radix Sort. I will show this with n.
3. Lastly We will sort these characters using Count sort which has a complexity of $\theta(m + k)$. Where n is the number of characters we will sort which is m. And the k is the range of the elements which is 27 since we have ”_” character. Therefore it will have a complexity of $\theta(m + 27) = \theta(m)$.

To sum up, Algorithm’s complexity is $\theta(m*n*(m+k))$. This is because we would Use count sort for m words for m elements for n characters. We can also simplify the equation using these calculations $\theta(m * n * (m + k))\theta(m * n * (m + 27)) = \theta(m * n * m) = \theta(m * n)$