

## Mid Term Exam

**Seneca**SCHOOL OF INFORMATION AND  
COMMUNICATIONS TECHNOLOGY**TERM COURSE NAME Professor COURSE CODE**

Fall 2023	Data Structures and Algorithms	Elham Ahmadi	DSA456
-----------	--------------------------------	--------------	--------

Name	Neron Parmar
Student Number	171690217
Section	V1C

DATE: Oct 18th, 2023

TIME ALLOWED: 2 hours

TOTAL MARKS 50

MARKS

As a Seneca student, you must conduct yourself in an honest and trustworthy manner in all aspects of your academic career. A dishonest attempt to obtain an academic advantage is considered an offense, and will not be tolerated by

**Question 1 (15 marks): Hash Tables**

Suppose you were given records with the following keys (the values are irrelevant to this discussion). The hash values for the keys are in the given table:

Key	Hash(key)
ash	17
bee	33
cow	55
data	18
eel	24
fly	46
glue	37
hip	30
ice	56
jam	19
kiwi	61

a) Using “**linear probing without tombstones**”, insert the following keys in the order given into the table below:

bee, cow, data, eel

			Bee			cow	eel		data
0	1	2	3	4	5	6	7	8	9

b) Now, insert that old list as “bee, cow, data, eel” into this table:

Cow	eel		bee	data
0	1	2	3	4

c) Suppose you were given the following table, what would the table look like if you were to remove **jam** make sure to cross out any key that is no longer where they use to be. Explain your answer to get full mark.

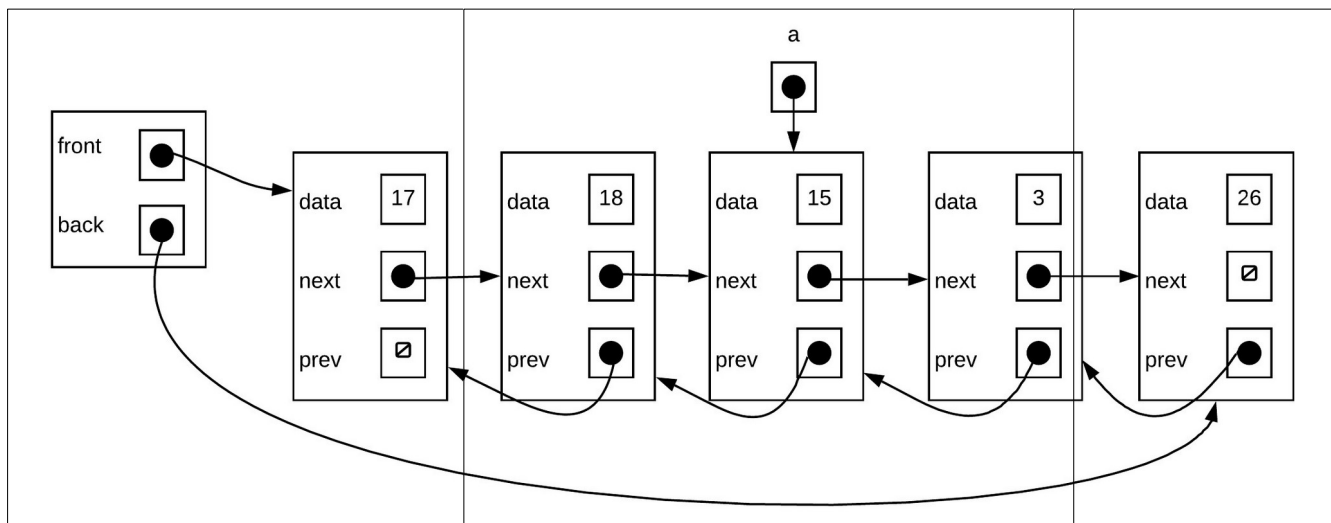
hip	glue	kiwi					bee				ash	data	jam
0	1			2			3	4	5	6	7	8	9
		hip	glue	kiwi	bee	ash	data						
		0	1	2	3	4	5	6	7	8			

**Answer C:**

In the given table of keys, "jam" was initially in slot 8 but has been removed. To maintain the order of the keys and close the gap, the other keys to the right of "jam" have been shifted to the left. After the shifting, "hip" remains in slot 0, "glue" in slot 1, "kiwi" in slot 2, and "bee" in slot 3. "Ash" and "data" that were originally in slots 5 and 6 respectively, have moved to slots 4 and 5 to fill the gap. The updated table shows the keys in their new positions or the ones that have been removed. The updated table is now consistent, and no gaps are left after the removal of "jam."

**Question 2 [10 marks]:**

The drawing below shows the current state of a linked list.



a) Alter the above diagram-as altering the connectivity between nodes to show what the list would look like if you were to perform the following code blurb on the list: (you can do the drawing on paint/word/hand writing whatever you prefer.)

```

a.prev.next = a.next
a.next.prev = a.prev
a.next = self.front
a.prev = None
self.front.prev = a

```

b) The above code aims at moving the node a points at (the one with 15) to the front of the list, but still there is one missing line code at the end of code. Write down that single line of code.

**Question 3 [10 marks]:**

Perform an analysis of each of the following functions with respect to number. You should provide mathematical analysis for each section.

```
def function1(number): i = 1
    rc = 0
    while(i < number):
        rc += 1
        i = i * 2
    return rc
```

```
def function2(number): i = 0
    rc = 0
    while (i < 5):
        rc += number
        i += 1
    return rc
```

```
def function3(number): i = 0
    rc = 0
    while(i < number):
        i += 1
        rc += 1
        number = number // 2    #(for example: 4//2=2, 10//3=3)
    return rc
```

**ANSWER 3:**

## Function 1:

This function uses a while loop to repeatedly multiply 2 with the initial value of  $i$  until it becomes greater than or equal to the given number. The function counts the number of iterations performed in the while loop and returns it.

## Analysis:

- The complexity of this function is logarithmic with respect to the input number.
- The while loop runs for  $\log_2(\text{number})$  times.
- Therefore, the time complexity of this function is  $O(\log n)$ .

## Function 2:

This function uses a while loop to repeat the same process five times. In each iteration, it adds the given number to the  $rc$  variable and increments the value of  $i$ . After five iterations, the function returns the value of  $rc$ .

## Analysis:

- The complexity of this function is constant with respect to the input number.
- The while loop runs for a fixed number of iterations (5 times) regardless of the input number.
- Therefore, the time complexity of this function is  $O(1)$ .

## Function 3:

This function uses a while loop to iterate until the value of  $i$  becomes equal to the given number. In each iteration, it increments the value of  $i$  and  $rc$ . Additionally, it updates the value of the given number by dividing it by 2. The function returns the value of  $rc$ .

## Analysis:

- The complexity of this function is linear with respect to the input number.
- The while loop runs for  $n$  times, where  $n$  is the value of the input number.
- Therefore, the time complexity of this function is  $O(n)$ .

Overall, the time complexity of function 1 is logarithmic  $O(\log n)$ , function 2 is constant  $O(1)$ , and function 3 is linear  $O(n)$  with respect to the input number.

**Question 4 [5 marks]:**

Analyze the following recursive function with respect to **number**:

```
def recursive (x, number):  
    if number==0:  
        return 0  
    elseif number==1:  
        return x  
    else:  
        return x+ recursive(x,number-1)
```

**Answer:**

In this function, I think that this is designed to calculate the product of two numbers: x and the number. There are two parameters passed in. It is assumed that the number parameter is a non-negative integer and that uses the recursive approach to perform the calculation. For the base case, if the number get equal to 0, then the function will return 0 while if the number gets equal to the 1, then the function will return x. While in the recursive case, if the values of the number are greater than 1 then the function will make a recursive call to itself with the number variable which gets decremented by 1.



Question 5 : 10 marks

Show step by step how 1) quick sort and 2) Selection sort works on array [8,10,15,3,13,9,8]?

Explain the details of each step. You can type here the answer, or attaching a picture of hand written solution

**Answer:**

The steps for the quick sort and the selection sort is as follows for the array list [8,10,15,3,13,9,8].

**FOR QUICK SORTING:**

For doing sorting using the quick sort these are the steps:

Step1:

- Taking the original array list that is [8, 10, 15, 3, 13, 9, 8]

Step2:

- Selecting the last element that is 8 as a pivot, partitioning the array by rearranging all the elements around the pivot in such a way that all the elements are less than the pivot and are placed to the left and all the elements that are greater than the pivot are placed on the right. So, after doing this step the array list will become like this,

Array list: [3, 8, 8, 10, 15, 13, 9].

Step3:

- Now, after do this, again by repeating the previous step, we divide the array in part in left and right part for the left part it is [3,8,8] and for the right part [10,15,13,9].
- So now taking the left part that is [3,8,8], choosing the last element 8 as the pivot and then partitioning the array, the result that will be obtained will be [3, 8, 8]
- Now same way selecting the right part that is [10, 15, 13, 9], selecting the last element that is 9 as the pivot and again performing partitioning the array, which will be resulting in [10,15,13,9]. So now the element 9 will be on the final sorted position.

Step4:

- Now again repeating the previous steps, there will be not two parts remaining they are, [3,8,8] and [10,15,13]. So again for the [3,8,8] part the pivot that is 8 is already on the sorted position which is its final result so this part is sorted
- Now for the unsorted part that is [10,15,13], choosing the last element number that is 13 as the pivot and partitioning it the array will be resulted like this, [13,13,15].
- So, after partitioning of the pivot element 13 is on its final position. So the sorted list will be ,  
Sorted list: [3, 8, 8, 9, 10, 13,15]

**FOR SELECTION SORTING:** In the selection sorting, first dividing the input array in two parts that is the sorted one and another is unsorted, later from the algorithm selecting the element from the unsorted part and then moving it to the sorted part. These are the steps below on doing it:

Step1:

- The initial array that is [8,10,15,3,13,9,8].

Step2:

- Finding the minimum element in the array list.
- That is basically the number 3 and after that swapping that number 3 to the first element place which is number 8. So, it will become somewhat like this after it gets swapped [3,10,15,8,13,9,8]. Then moving to step 3.

## Step3:

- Again finding the minimum element in the array list, but in the unsorted part of the array list, however here the 3 number element which is on the first place should not be counted as it is sorted.
- So, now the element which is minimum is 8 so it will come at the second place in the list. So, we will swap it with 10 number to bring it to the second place. Moreover, there are two 8 number elements. So, the number 8 will be swapped with two number they are with 10 and 15. So after sorting it will be somewhat like this;

Sorted list: [3, 8, 8, 10, 13, 9, 15]

## Step4:

- Again by taking the minimum element, this time it will be 9.
- So 9 will be swapped with the number 10. So the sorted list will be like: [3, 8, 8, 9, 13, 10, 15].

## Step5:

- Now again the smallest element from the unsorted part will be selected that is the number 10. Similarly, the unsorted elements will keep swapping until the list gets sorted after the number 10, it will select the number 13 and after that 15. So, the whole sorted or the fully sorted list will be like this,

Fully sorted list = [3, 8, 8, 9, 10, 13, 15]