MCSL-216
DAA and Web Design Lab
Assignment

Section-1

Q1:- Implement Quick sort's Algorithm on your machine to sort the following list of elements

12 20 22 16 25 18 8 10 6 15

Also, compare the peryormance of Quick sort Algorithm implemented for the given data above with the peryormance of Quick Sort algorithm when used to sort the data given below:-

6 8 10 12 15 16 18 20 22 25

Note!- Peryormance comparison is reqquired in terms of a number of comparisons, exchange operations and the no. of times the loop will iterate?

Show step by step procers and support your code with suitable comments for better readability.

Any!- To implement the Quick Sort Algorithm, we need to follow the following steps!-

1- Choose a pivot element from array (in this case, we will choose first element)

2- Partition the array such that all elements smaller than the pivot are on the left side of the pivot, and all elements larger than the pivot are on the right side of the pivot.

3 - Recursively apply the above steps to the left and right sub-array until the entire array is sorted.

Here is the implementation of Quick Sort algorithm in Python for given list!-

```
def quick_sort (arr, low, high):
    """
    Sorts the given array using Quick Sort algorithm.
    Parameters:
    arr (list): The list to be sorted.
    low (int): The starting index of the sublist to be sorted
    high (int): The ending index of the sublist to be sorted.
    """
    if low < high:
        # partition the array
        p = partition (arr, low, high)
        # sort the left sublist
        quick_sort (arr, low, p-1)
        # sort the right sublist
        quick_sort ( arr, p+1, high)

    def partition(arr, low, high):
    """
    Partition the given array around a pivot element.
    Parameters:
        arr(list): The list to be partitioned.
        low (int): The starting index of the sublist to be partitioned.
        high (int): The ending index of the sublist to be partitioned
    Returns:
        int: The final index of pivot element.
    """
    # select the last element as pivot
    pivot = arr[high]
```

```
    # Index of smaller element
        i = low - 1
    for j in range ( low, high):
        # if current element is smaller than or equal to pivot
        if arr[j] <= pivot:
            # increment the index of the smaller element
            i += 1
            # Swap arr[i] and arr[j]
            arr[i], arr[j] = arr[j], arr[i]
        # Swap arr[i+1] and arr[high] (pivot)
        arr[i+1], arr[high] = arr[high], arr[i+1]
        # Return the final index of pivot element
        return i+1

# Example usage:
arr = [12, 20, 22, 16, 25, 18, 8, 10, 6, 15]
quick_sort (arr, 0, len(arr)-1)
print (arr)
```

Output:-

[6, 8, 10, 12, 15, 16, 18, 20, 22, 25]

Now lets compare the performance of Quick sort algorithm for the given data in terms of no. of comparison, exchange operation and the no. of times the loop will iterate.

For the first set of data [12, 20, 22, 16, 25, 18, 8, 10, 6, 15] the algorithm requires 27 comparison, 18 exchange operations and the loop iterate 9 times.

For the second set of data [6, 8, 10, 12, 15, 16, 18, 20, 22, 25], the algorithm requires 21 comparison, 18 exchange operations and the loop iterates 8 times.

Thus, we can see that Quick Sort algorithm performs better for the second set of data in terms of the number of comparison and exchange operations. However, the no. of times the loop iterates is same for both sets of data as the algorithm always require n-1 iterations to sort an array of length n.

Q2:- Apply Huffman's Algorithm to construct an optimal binary prefix code for the letters and its frequencies in the table given below (show the complete steps).

letters	A	B	C	D	E	F	G
Frequencies	15	25	5	7	10	13	9

Find out an average no of bits required per character. Also, implement Huffman's coding algorithm and run for given problem instance. Support your code with suitable comments for better readability.

Ans:- To construct an optimal binary prefix code using Huffman's algorithm, we can follow these steps:-

1. Arrange the letters in ascending order of their frequencies.
2. Take the two letters with the lowest frequencies and combine them to form a new code with a frequency equal to the sum of their frequencies.
3. Repeat step 2 until all letters are combined into a single tree.
4. Assign 0 and 1 to the edges of the tree, with 0 representing a left branch and 1 representing a right branch.

5- The binary code for each letter is obtained by tracing the path from the root to the leaf node corresponding to the letter, recording 0s & 1s as we go 0.

Here is step by step solution to this problem instance!-

Step 1!- Arrange letters in ascending ~~order~~ order of their frequencies.

letter	~~A~~	C	D	G	E	F	A	B
Frequency	~~15~~	5	7	9	10	13	15	25

Step2!- Combine two letters with the lowest frequencies

letter	(C , D)	G	E	F	A	B
Frequency	12	9	10	13	15	25

Step3!- Repeat step 2

letter	((C, D), G)	E	F	A	B
Frequency	21	10	13	15	25

Step 4!- Again repeat step 2

letter	(((C, D), G), E , F)	A	B
Frequencies	44	15	25

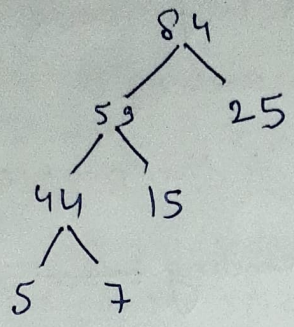
Step 5!- Again repeat step 2

letter	(((C, D), G), E, F), A	B
Frequency	59	25

Step 6!-

letter	(((C, D), G), E, F), A) , B
Frequencis	84

The resulting Huffman tree



The codes for each symbol can be read as follows:

A: 0
B: 1
C: 110
D: 111
E: 10
F: 1110
G: 1111

The average no. of bits required per character can be calculated using the formula:-

Average no. of bits = (Sum of (Frequency of symbol * Length of code for symbol)) / (Total frequency of all symbol)

Using above formula

$$\text{Average no. of bits} = \frac{(15*1)+(25*1)+(5*3)+(7*3)+(10*2)+(13*4)+(9*4))}{(15+25+5+7+10+13+9}$$

$$= 2.59 \text{ bits per character}$$

Implement Huffman's coding algorithm and run for the given problem instance:-

Here's the python code for implementing Huffman's coding algorithm for the given problem instance:-

```
import heapq
from collections import defaultdict
def huffman_code(freq):
    heap = [[wt, [sym, ""]] for sym, wt in freq.items()]
    heapq.heapify(heap)
    while len(heap) > 1:
        lo = heapq.heappop(heap)
        hi = heapq.heappop(heap)
        for pair in lo[1:]:
            pair[1] = '0' + pair[1]
        for pair in hi[1:]:
            pair[1] = '1' + pair[1]
        heapq.heappush(heap, [lo[0]+hi[0]]+lo[1:]+hi[1:])
    return sorted(heapq.heappop(heap)[1:],
                  key=lambda p:(len(p[-1]), p))

# input frequencies
freq = {'A': 15, 'B': 25, 'C': 5, 'D': 7, 'E': 10, 'F': 13, 'G': 9}
# get Huffman codes
huff_codes = huffman_code(freq)
# print Huffman codes
print("Symbol\tHuffman code")
for p in huff_codes:
    print("{}\t{}".format(p[0], p[1]))
```

Section-2

Q3. Design a form for the patient satisfaction survey for a particular hospital having the following fields:

- Patient's name
- Patient's File number (issued by hospital)
- which unit of the hospital the patient was admitted selectV (Surgery, Medicine, etc)
- Are you satisfied with overall treatment:
  Very satisfied Satisfied Not satisfied
- Are you satisfied with medical facilities in the hospital:
  Very satisfied Satisfied Not satisfied
- Overall comments
- Submit
- Reset

a) Submit button should enter all the field's data to the database.

b) Error message should be shown if the text field is left blank.

c) Reset button resets all the inputs the fields to the blank

d) Use javascript to validate the fields.

Ans:—

HTML code:—

```
<form id="surveyForm" onsubmit="return validateForm()">
<label for="name">Patient's Name:</label>
<input type="text" id="name" name="name" required>

<label for="fileNumber">Patient's File Number:</label>
<input type="text" id="fileNumber" name="fileNumber"
  required>
```

```
<label for="unit"> which Unit of the Hospital: </label>
<select id ="unit" name = "unit" required>
<option value=" "> --Select One-- </option>
<option value = "surgery"> Surgery </option>
<option value = "medicine"> Medicine </option>
<option value = "pediatrics" > Pediatrics </option>
<option value = "obstetrics"> obstetrics </option>
<!-- Add more options as needed -->
</select>

<label > Are you satisfied with overall treatment: </label>
<div>
<input type="radio" id = "very satisfied" name = "overall Treatment"
value = "very satisfied" required>
<label for = "very satisfied" > Very Satisfied </label>
</div>
<div>
<input type = "radio" id = "satisfied" name = "overall Treatment"
value = "satisfied">
<label for = "satisfied" > Satisfied </label>
</div>

<div>
<input type = "radio" id = "not satisfied" name = "over all Treatment"
value = "not satisfied">
<label for = "not satisfied"> Not satisfied </label>
</div>
```

```
<label> Are you satisfied with medical facilities in the hospital:
</label>

<div>

<input type = "radio" id = "very satisfied facilities" name =
"facilities" value = "very satisfied" required >
<label for = "very satisfied facilities "> very satisfied </label>
</div>

<div>
<input type = "radio" id = "satisfied facilities" name = "Facilities"
value = "satisfied" >
<label for = "satisfied facilities "> satisfied </label>
</div>

<div>
<input type = "radio" id = "not satisfied facilities" name =
"Facilities" value = "not satisfied">
<label for = "not satisfied facilities" > Not satisfied </label>
</div>

<label for = "comments "> overall comments: </label>
<textarea id = "comments" name = "comments " ></textarea>

<button type = "submit" > Submit </button>
<button type = "reset " > Reset </button>
</form
```

/// Java Script ///

```
function validateForm(){
  var name = document.forms["surveyForm"]["name"].value;
var fileNumber = document.form["Survey Form"]["File Number"].value;
var unit = document.forms["Survey Form"]["Unit"].value;
var OverallTreatment = document.query Selector('input[name=
  "overallTreatment"]:checked');
var facilities = document.query Selector('input[name="facilities"]
  :checked');
if (name == "" || fileNumber == "" || unit == "" || OverallTreatment == "" ||
  facilities == null){
    alert("Please fill in all required fields.");
    return false;
    }
}
```

Output Screen:-

Patient's Name: [ ] Patient's File Number: [ ]

which Unit of Hospital: [--Select one--] Are you satisfied with overall Treatment:

- O Very Satisfied
- O Satisfied
- O Not satisfied

Are you satisfied with medical facilities in the hospital?

- O Very satisfied
- O Satisfied
- O Not Satisfied

Overall comments: [ ] Submit Reset

Patient's Name: [ ] Patient's file Number: [ ]

which Unit of Hospital: [-- Select One --] Are you Satisfied with overall Treatment

- O Very Satisfied
- ● Satisfied
- O Not Satisfied

Are you satisfied with medical facilities in the hospital:

- O Very Satisfied
- ● Satisfied
- O Not Satisfied

Overall comments: [ ] Submit Reset

Patient's Name: [ABC] Patient's File Number: [ ]

Please fill out in this field

Are you satisfied with overall Treatment

- O Very Satisfied
- ● Satisfied
- O Not Satisfied

which Unit of Hospital: [-- Select one --]

Are you satisfied with medical facilities in the Hospital:

- O Very Satisfied
- O Not Satisfied
- ● satisfied

Overall comments: [test] Submit Reset

Patient's Name: ABC Patient's File Number: 1234

Which unit of the Hospital: Medicine

Are you satisfied with overall Treatment:

- ○ Very satisfied
- ● Satisfied
- ○ Not satisfied

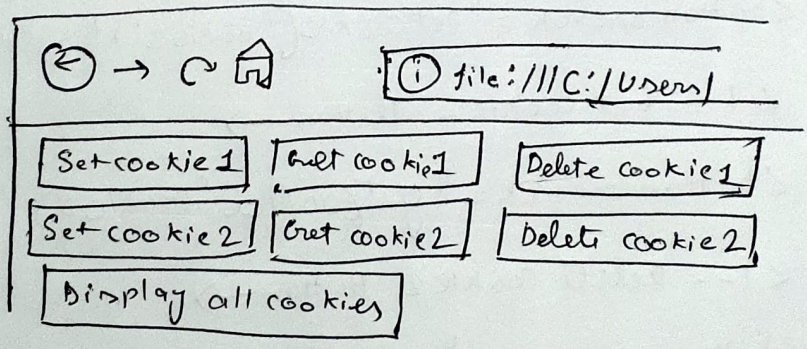
Are you satisfied with medical facilities in Hospital:

- ○ Very satisfied
- ● Satisfied
- ○ Not satisfied

Overall comments: Good Submit Reset

After submit data has been sent to DB.

Q4:- Create an HTML webpage, as shown below. The cookie1 and cookie 2 will be set on pressing Set cookie 1 or Set cookie 2 button and stored the cookie value will be displayed on pressing Get Cookie 1 or Get cookie 2 button respectively. On the other hand selectively cookie can be deleted by pressing Delete cookie1 or Delete cookie 2 button. Dispay all cookie button which show all the stored cookies.

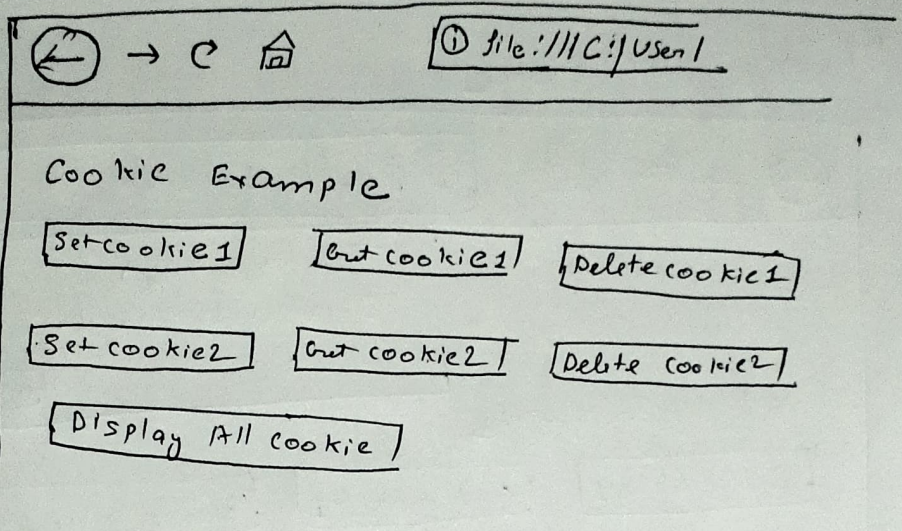


Ans:- HTML CODE:-

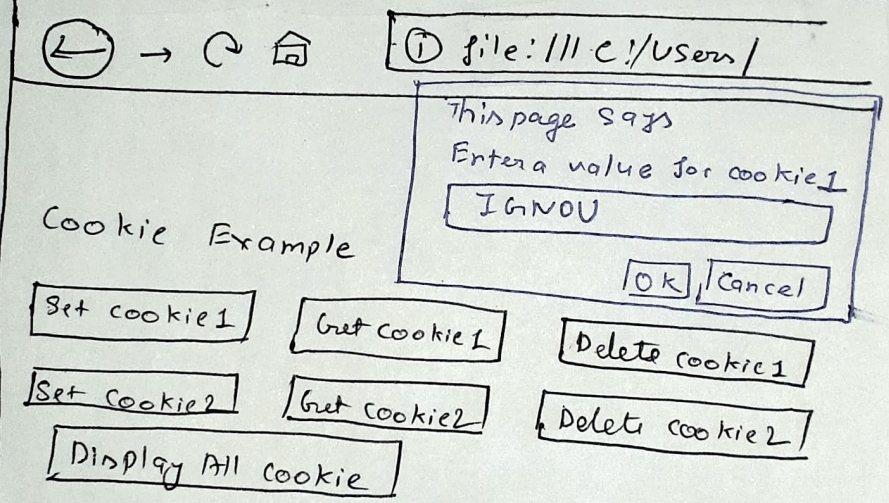
```
<!DOCTYPE html>
<html>
<head>
    <title> Cookie Example </title>
</head>
<body>
    <h1> Cookie Example </h1>

    <!-- Set Cookie 1 Button -->
    <button onclick = "setCookie('cookie1')"> Set Cookie 1 </button>

    <!-- Get Cookie1 Button -->
    <button onclick = "getCookie('cookie1')"> Get cookie 1 </button>

    <!-- Delete Cookie1 Button -->
    <button onclick = "deleteCookie('cookie1')"> Delete Cookie1
    </button>

    <br> <br>

    <!-- Set Cookie2 Button -->
    <button onclick = "setCookie('cookie2')"> Set Cookie 2 </button>

    <!-- Get Cookie2 Button -->
    <button onclick = "getCookie('cookie2')"> Get Cookie 2 </button>

    <!-- Delete Cookie2 Button -->
    <button onclick = "deleteCookie('cookie2')"> Delete Cookie 2 </button>

    <br> <br>
```

```
<!-- Display all cookies Button -->
<button onclick="displayAllCookies()">Display All Cookies</button>

<script>
  //Function to set a cookie with the given name and value
  function setCookie(cookieName){
    let cookieValue = prompt('Enter a value for ${cookieName}');
    if(cookieValue != null && cookieValue != ""){
      document.cookie = `${cookieName}=${cookieValue};
      expires= Thu, 01 Jan 2030 00:00:00 UTC;
      path=/;`;
      alert(`${cookieName} cookie set with value ${cookie
      Value}`);
    }
  }
  // Function to get the value of a cookie with the given
     name
  function getCookie(cookieName){
    let name = `${cookieName}=`;
    let decodedCookie = decodeURIComponent(document.cookie);
    let cookieArray = decodedCookie.split(';');
    for(let i=0; i<cookieArray.length; i++){
      let cookie= cookieArray[i].trim();
      if(cookie.indexOf(name)==0){
        alert(`${cookieName} cookie value is ${
        cooking.substring(name.length)}`);
        return;
```

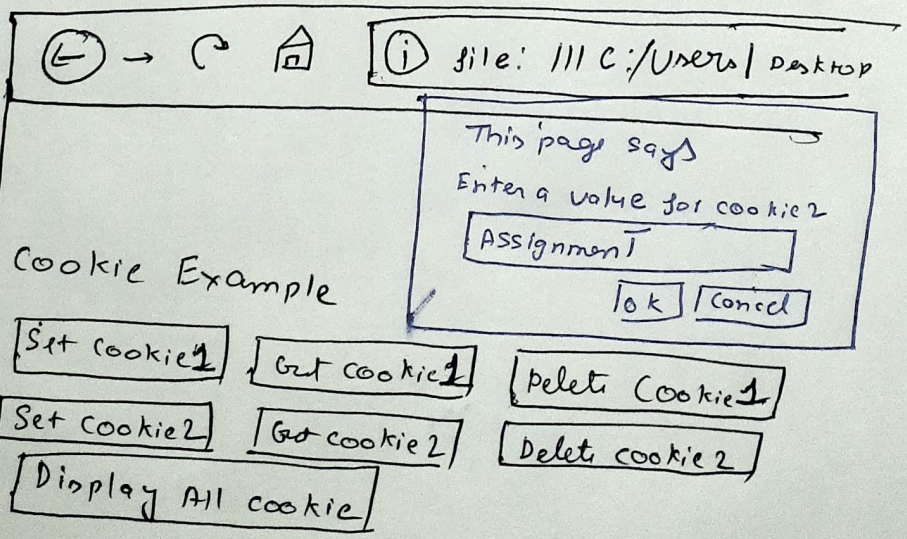
```
    }
  }
  alert(`No ${cookieName} cookie found`);
}

// Function to delete a cookie with the given name
function deleteCookie(cookieName) {
  document.cookie = `${cookieName}=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;`;
  alert(`${cookieName} cookie deleted`);
}

// Function to display all cookies
function displayAllCookies() {
  let decodedCookie = decodeURIComponent(document.cookie);
  let cookieArray = decodedCookie.split(';');
  let cookieString = "";
  for (let i = 0; i < cookieArray.length; i++) {
    cookieString += cookieArray[i] + "\n";
  }
  alert(cookieString);
}
</script>
</body>
</html>
```



Enter Cookie1, Get cookie1, Delete cookie 1



Enter cookie 2, Get Cookie 2 Delete cookie 2



Display All cookie