

# ASSIGNMENT REPORT- 2

(Task : *to solve all the given problem*)

Submitted by

(Mogalraj Kushal Dath)

Registration Number : 12100559

Course Name : Advanced Data Structure and Algorithms

Course Code : CSE523

Section Number : K21ML

Roll Number : RK21MLA01

Submitted to

(Tejinder Thind Sir)

School of Computer Science and  
Engineering



**L** LOVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

1) Sol: Given String to be Encoded is given by

"AABABACBAA CBAA DAAA..."

→ Inputs  $l_1 = 4$  and  $l_2 = 4$  also known as Lookahead buffer and search buffer respectively.

→ Code word  $\Rightarrow \langle [1], [2], [3] \rangle$

[1] Position in which match is found.

[2] Length of the match.

[3] First Mismatching Symbol.

Input / Message	Buffer ( $l_1, l_2$ )	Code Transmitted																
A A B A B A C B A A C B A A D A A A ...	<table><tr><td>A</td><td>A</td><td>A</td><td>A</td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table>	A	A	A	A					0	1	2	3	4	5	6	7	$\langle 0, 0, A \rangle$
A	A	A	A															
0	1	2	3	4	5	6	7											
A A B A B A C B A A C B A A D A A A ...	<table><tr><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>B</td><td>A</td></tr><tr><td colspan="4"></td><td colspan="4">↑</td></tr></table>	A	A	A	A	A	A	B	A					↑				$\langle 2, 2, B \rangle$
A	A	A	A	A	A	B	A											
				↑														
A B A C B A A C B A A D A A A ...	<table><tr><td>A</td><td>A</td><td>A</td><td>B</td><td>A</td><td>B</td><td>A</td><td>C</td></tr><tr><td colspan="4"></td><td colspan="4">↑</td></tr></table>	A	A	A	B	A	B	A	C					↑				$\langle 2, 3, C \rangle$
A	A	A	B	A	B	A	C											
				↑														
B A A C B A A D A A A ...	<table><tr><td>A</td><td>B</td><td>A</td><td>C</td><td>B</td><td>A</td><td>A</td><td>C</td></tr><tr><td colspan="4"></td><td colspan="4">↑</td></tr></table>	A	B	A	C	B	A	A	C					↑				$\langle 1, 2, A \rangle$
A	B	A	C	B	A	A	C											
				↑														
C B A A D A A A ...	<table><tr><td>C</td><td>B</td><td>A</td><td>A</td><td>B</td><td>A</td><td>A</td><td>A</td></tr><tr><td colspan="4"></td><td colspan="4">↑</td></tr></table>	C	B	A	A	B	A	A	A					↑				$\langle 0, 3, A \rangle$
C	B	A	A	B	A	A	A											
				↑														
D A A A ...	<table><tr><td>C</td><td>B</td><td>A</td><td>A</td><td>D</td><td>A</td><td>A</td><td>A</td></tr><tr><td colspan="4"></td><td colspan="4">↑</td></tr></table>	C	B	A	A	D	A	A	A					↑				$\langle 3, 0, D \rangle$
C	B	A	A	D	A	A	A											
				↑														
A A A ...	...	—																

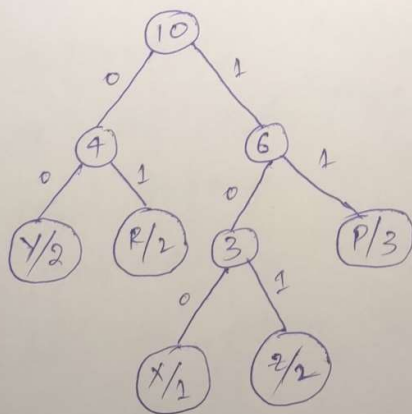
2) Q: Given Message/string to be Encoded using Huffman encoding technique is "XYVZZRPPPP"

→ Given that if two letters have same frequency then the letter which occurs earlier according to the dictionary order whose code length should be atmost the length of the code assigned to the other letter.

i.e,  $\boxed{\text{code length}(R) \leq \text{code length}(Y) \leq \text{code length}(Z)}$

Letter	frequency	Variable length code	Code length	Total Bits
X	1	100	3	3
Y	2	00	(2)	4
Z	2	101	(3)	6
R	2	01	(2)	4
P	3	11	2	6
				Total = 23

Code tree :



∴ the minimum length of the encoded string which satisfy both the properties is  
= 23 //

3) Q1:

- In general layman's term, Garbage collection is nothing but collecting (or) gaining memory back which has been allocated to objects but which is not currently in use in any part of our program.
- Garbage collection (GC) implemented differently for every language. Some high-level programming languages have some sort of built-in garbage collection. But for low-level programming languages may add garbage collection through libraries like in C-programming we use `malloc()` for memory allocation and `dealloc()` for memory deallocation.
- Let us take an example (or) program based on javascript to understand GC. javascript is a high-level language so we do not have to allocate memory manually. Memory allocation and releasing happens automatically. Making the memory free is the process of GC and there is a routine who does it called Garbage Collector.
- The Garbage collection considers references and it tries to release the memory if a location is not reachable.

### Program / Example :-

```
class useless {  
    public static void main (String[] args) {  
        obj = { name : 'Kushal' };  
        obj1 = obj;  
        obj = Null;  
        obj2 = null; } }
```

→ For this above example, I have created an object "obj" with the property 'name' as 'Kushal'. then I assigned "obj2" as well to this property.

→ If in case if I give ~~the~~ make these two objects (obj & obj2) references to Null then the memory where the value of the name is stored become useless i.e., no longer reachable then this location will be removed by the Garbage Collector.

→ There is algorithm called "Mark-and-sweep" which is used for the Garbage collection. This algorithm starts from root object and checks the references linked. if in case, this algorithm find unreachable locations then they are gonna be removed. All the modern browsers use this algorithm.