These filters are known as sobel filters and is used in edge detection.

Q4(a) Gaussian Noise also known as normal noise model is used frequently in practice.

The PDF of a Gaussian random variable z is given by

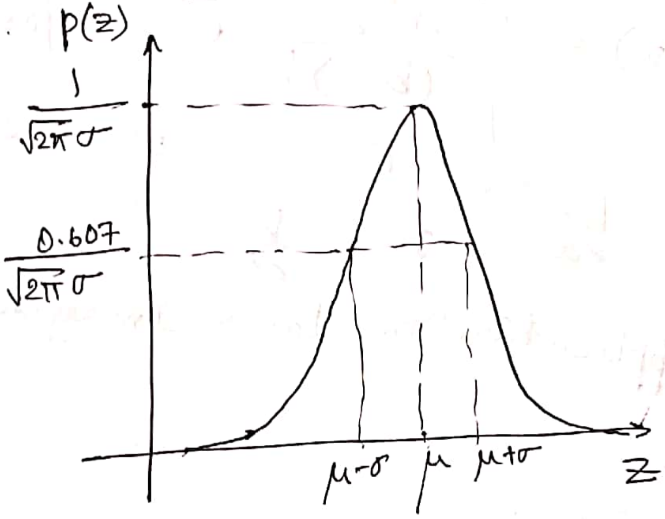
$$p(z) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

z = gray level.

μ = mean of average value of z.

σ = standard deviation of z.

In this distribution about 70% of its values will be in the range $[(\mu-\sigma), (\mu+\sigma)]$, and 95% in the range $[(\mu-2\sigma), (\mu+2\sigma)]$.



Anuran Chakraborty

Scanned with CamScanner

Roll : 001610501020

- Rayleigh Noise

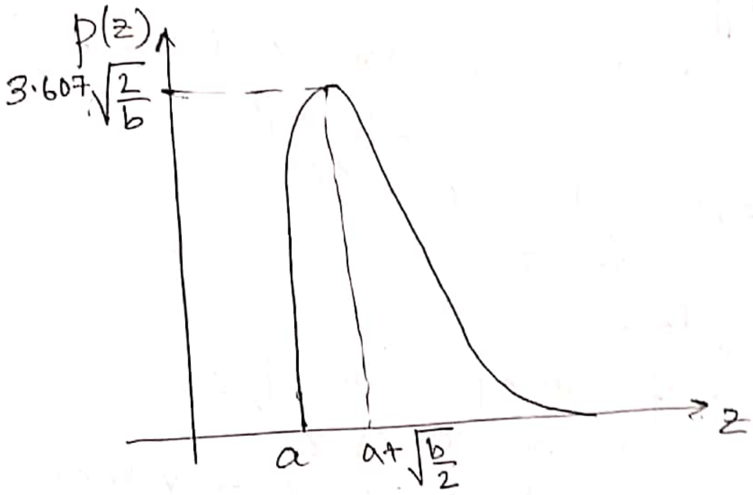
The PDF of the Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)\, e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a. \end{cases}$$

$$\mu = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}.$$

The shape is skewed to the right



- Gamma Noise This noise also known as Erlang noise has the PDF

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq a. \\ 0 & \text{for } z < a. \end{cases}$$

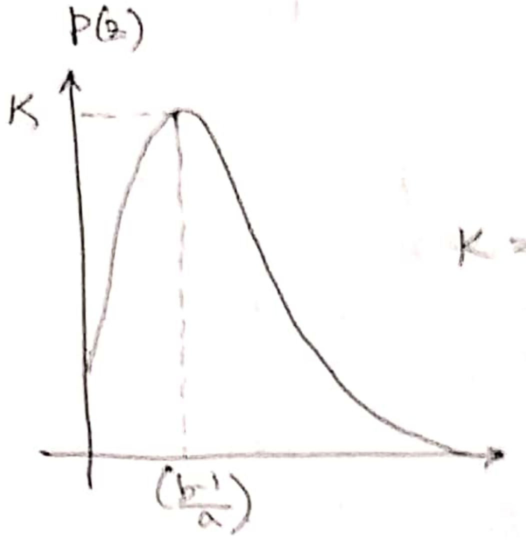
$$\mu = b/a \qquad \sigma^2 = \frac{b}{a^2}$$ ~~$a > 0$~~ [illegible].

$a > 0$, $b =$ +ve integer

It has application in Laser Imaging.

Anuran Chakraborty

Scanned with CamScanner



$$K = \frac{a(b-1)^{b-1}}{(b-1)!}e^{-(b-1)}.$$

Q4(b). Harmonic Mean filter.

It is used for removing ~~salt and pepper~~ noise

It is given by the equation

$$\hat{f}(x,y) = \frac{mn}{\displaystyle\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

where S_{xy} represents the set of coordinates in a rectangular subimage window of size m x n centered at the point (x,y).

$g(x,y)$ = pixel value of ~~at~~ corrupted image at coordinates (x,y)

$\hat{f}(x,y)$ = value of restored image at coordinates (x,y).

The harmonic mean filter works well for salt noise but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

Anuran Chakraborty

Scanned with CamScanner

Roll : 001610501020

Contraharmonic Mean filter

The contraharmonic mean filtering operation yields a restored image based on

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

where Q is called the order of the filter All other symbols have the same meaning as in the previous equation.

This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q is eliminates ~~salt noise~~ pepper noise. For negative values Q it eliminates salt noise. It cannot do both simultaneously.

if $Q=0$ it reduces to arithmetic mean filter

if $Q=-1$ it reduces to the harmonic mean filter.

Anuran Chakraborty

Scanned with CamScanner

Roll: 001610501020

Q5. Algorithm for construction of a Huffman Tree.

```
1. Begin
2. Create a table to store the frequency of each
   ~~character~~ symbol.
3. Create a leaf node for each symbol and, add it
   to the priority queue. Lower the probability
   higher the priority. (higher frequency means
   lower priority).
4. While there is more than one node in the
   queue:
     4.1 Remove the two nodes of highest priority
         from the queue
     4.2 Create a new internal node with these
         two nodes ~~and~~ as children and with
         probability equal to the sum of two
         nodes' probabilities.
     4.3 Add the new node to the queue
5. The remaining node is the root node and
   the tree is complete.
6. End.
```

Anuran Chakraborty

Scanned with CamScanner

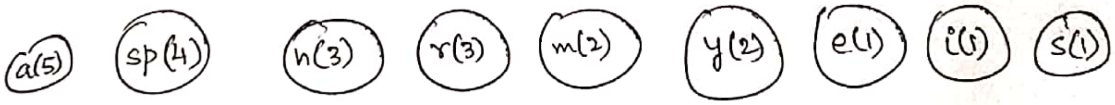
Roll:001610501020

my name is anuran chakraborty.

Frequency table

ch	Frequency
a	5
space	4
n	3
r	3
m	2
y	2
e	1
i	1
s	1
u	1
c	1
h	1
k	1
b	1
o	1
t	1

If the frequencies are same the characters are selected in ~~the~~ any order



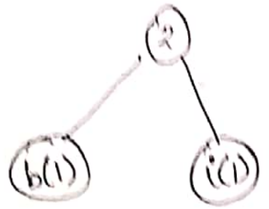
Anuran Chakraborty

Scanned with CamScanner

Roll: 001610501020.

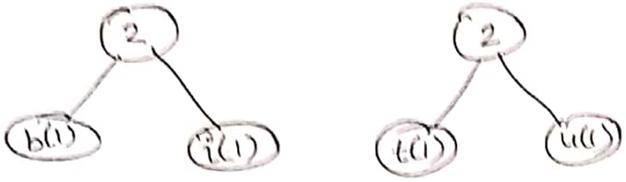
1. Select highest priority characters.

Choose b, i

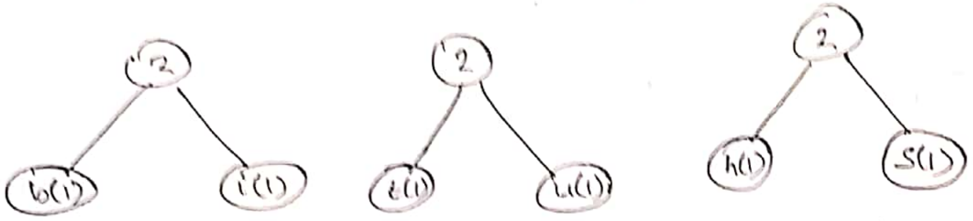


Remaining

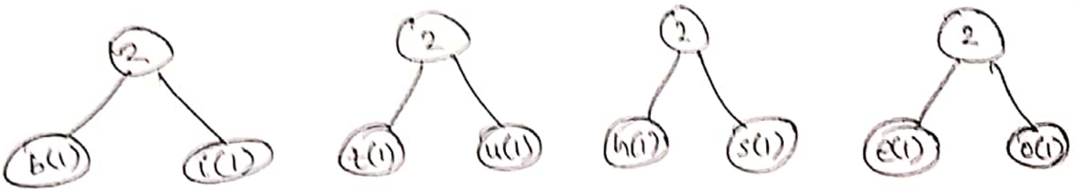
Choose t, u.



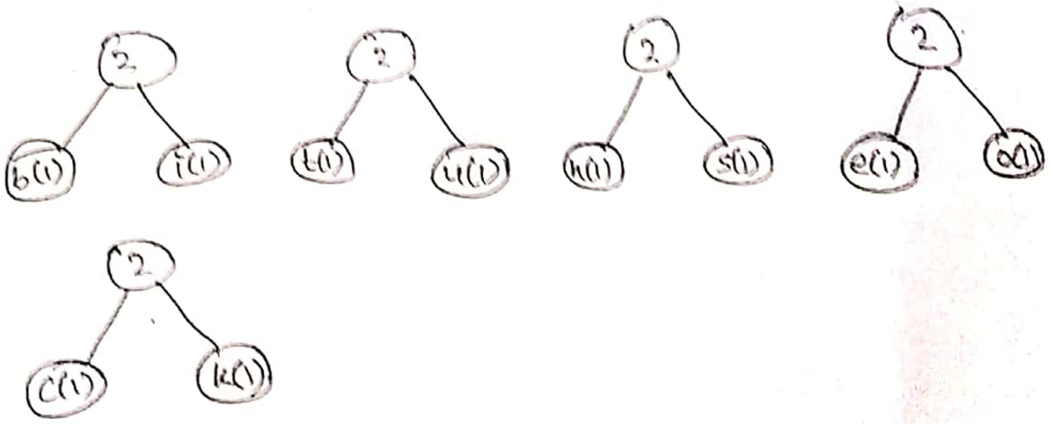
choose h, s.



choose e, o.



Choose c, k

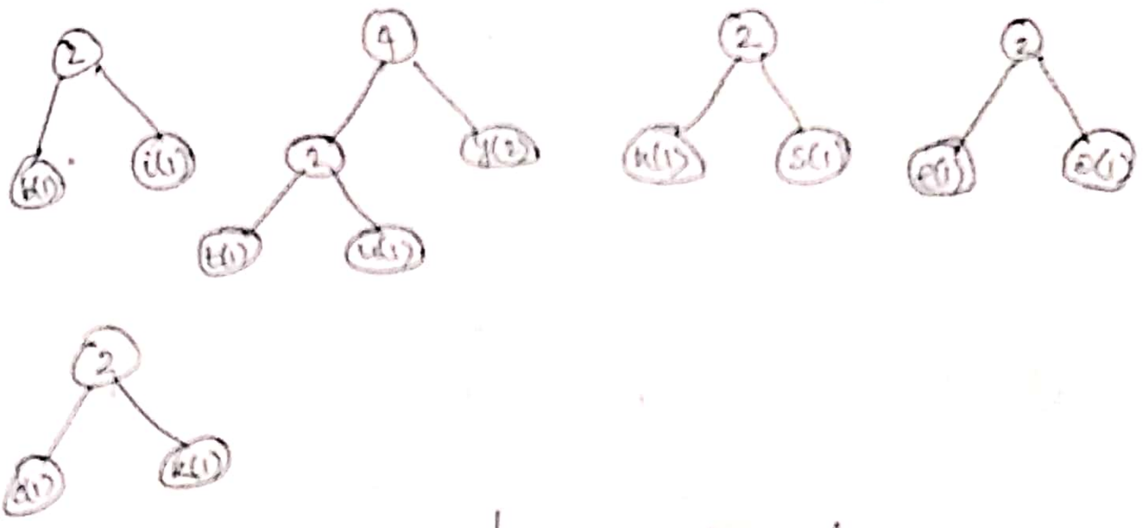


Anuran Chakraborty

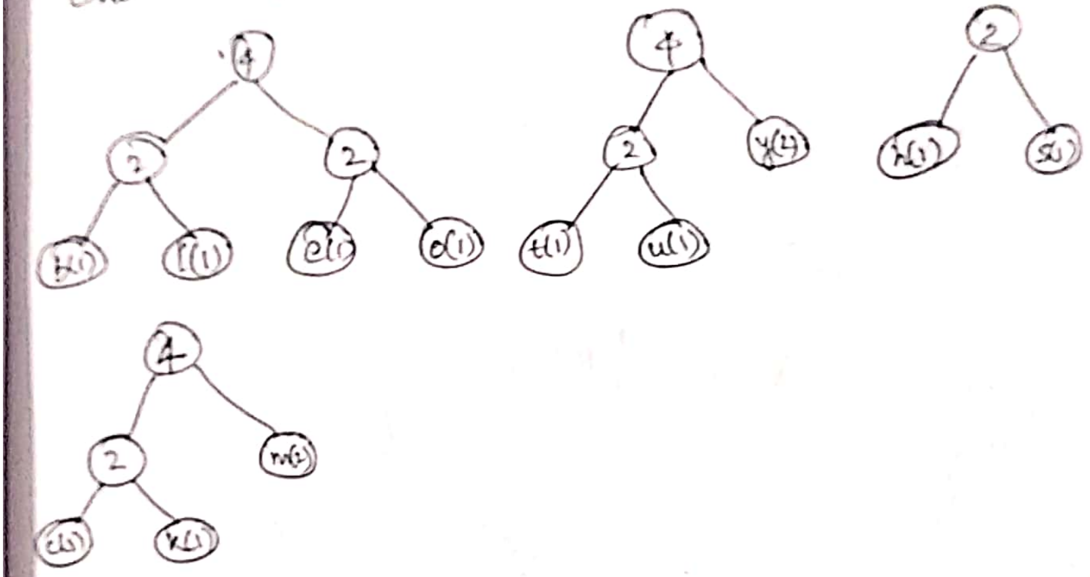
Scanned with CamScanner

201610501020

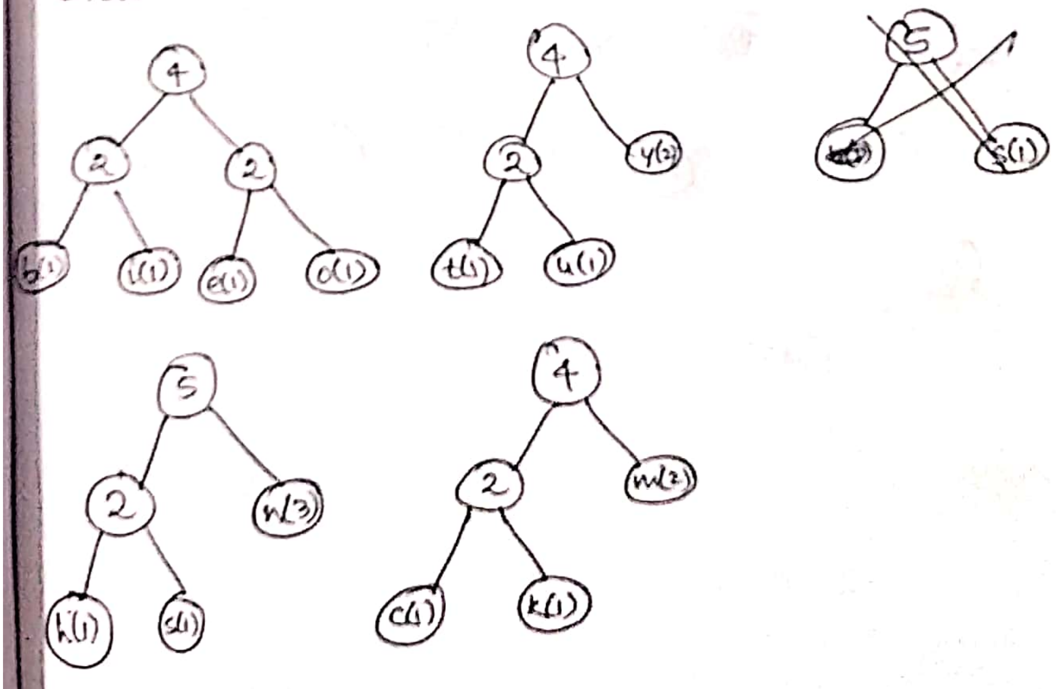
choose t, u and y as both have freq 2



choose b, i and e, o | and c, k and m



Choose h, s and n.

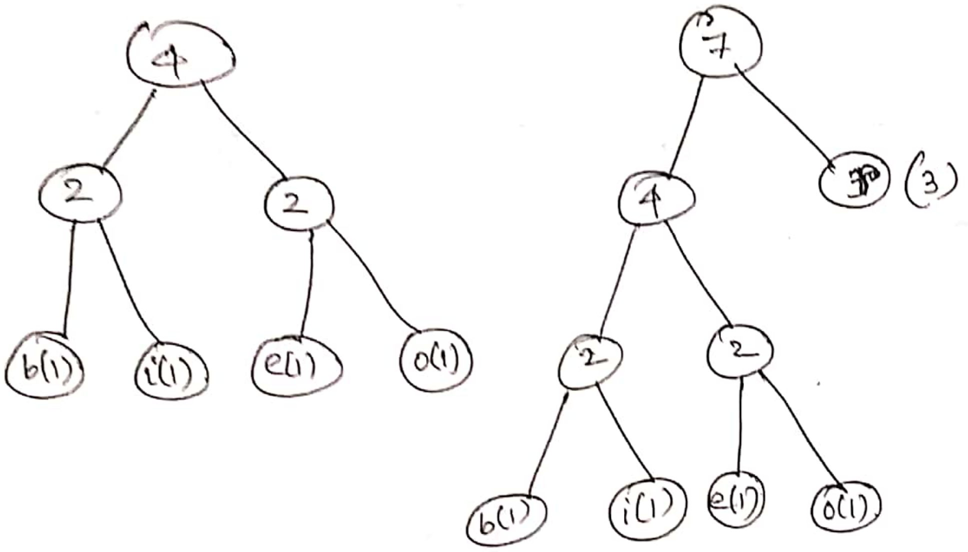


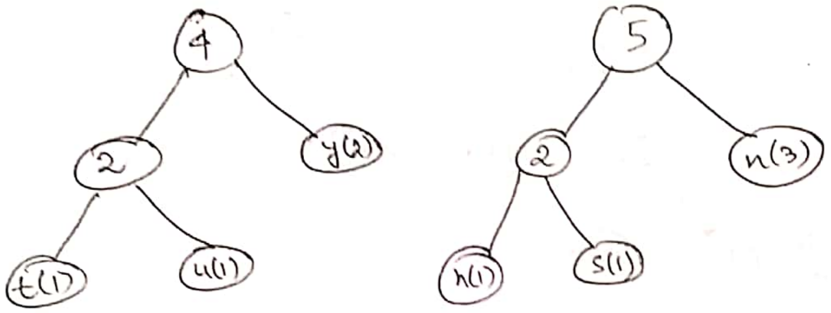
Anuran Chakraborty

Scanned with CamScanner

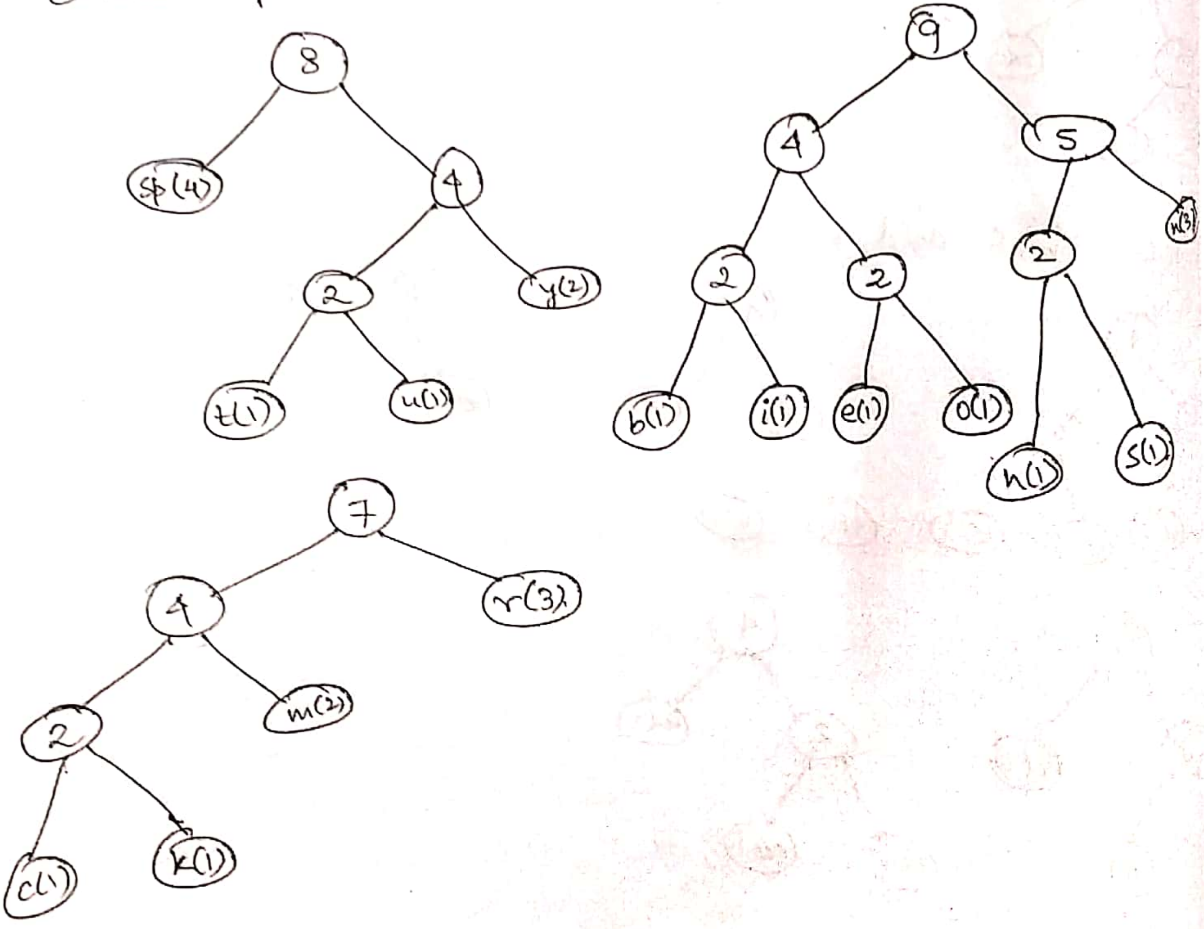
Roll: 001610501020

Next choose c,k,m and r.





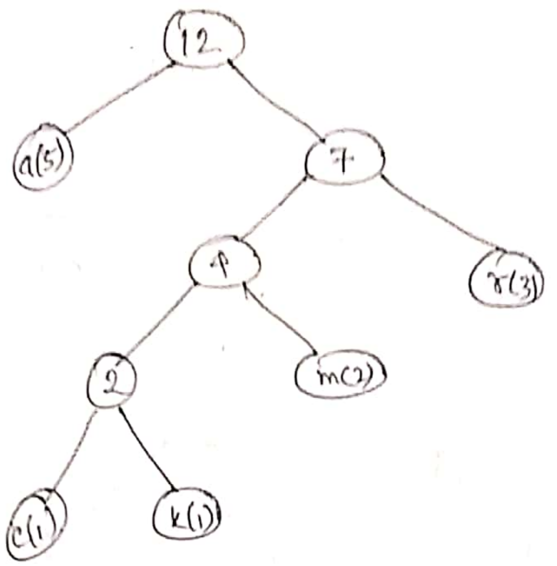
Choose space and t, y, u | Merge b,i,e,o and h,s,n

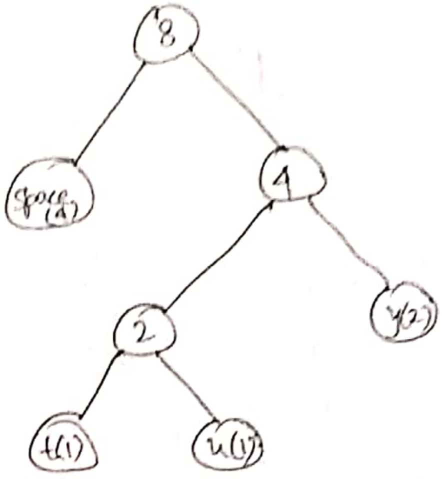


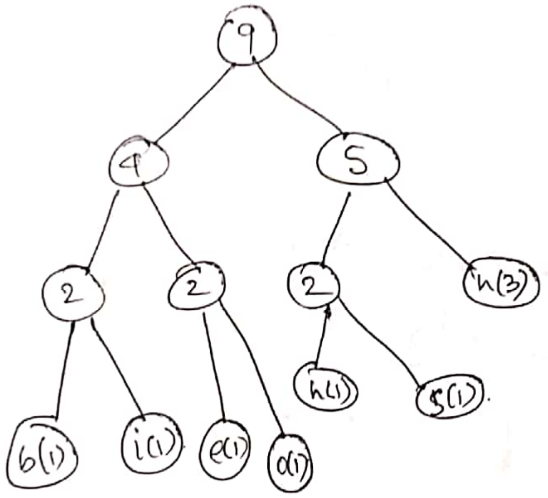
Anuran Chakraborty

Scanned with CamScanner

Choose ~~Merge~~ a(5) with c,k,m,r.

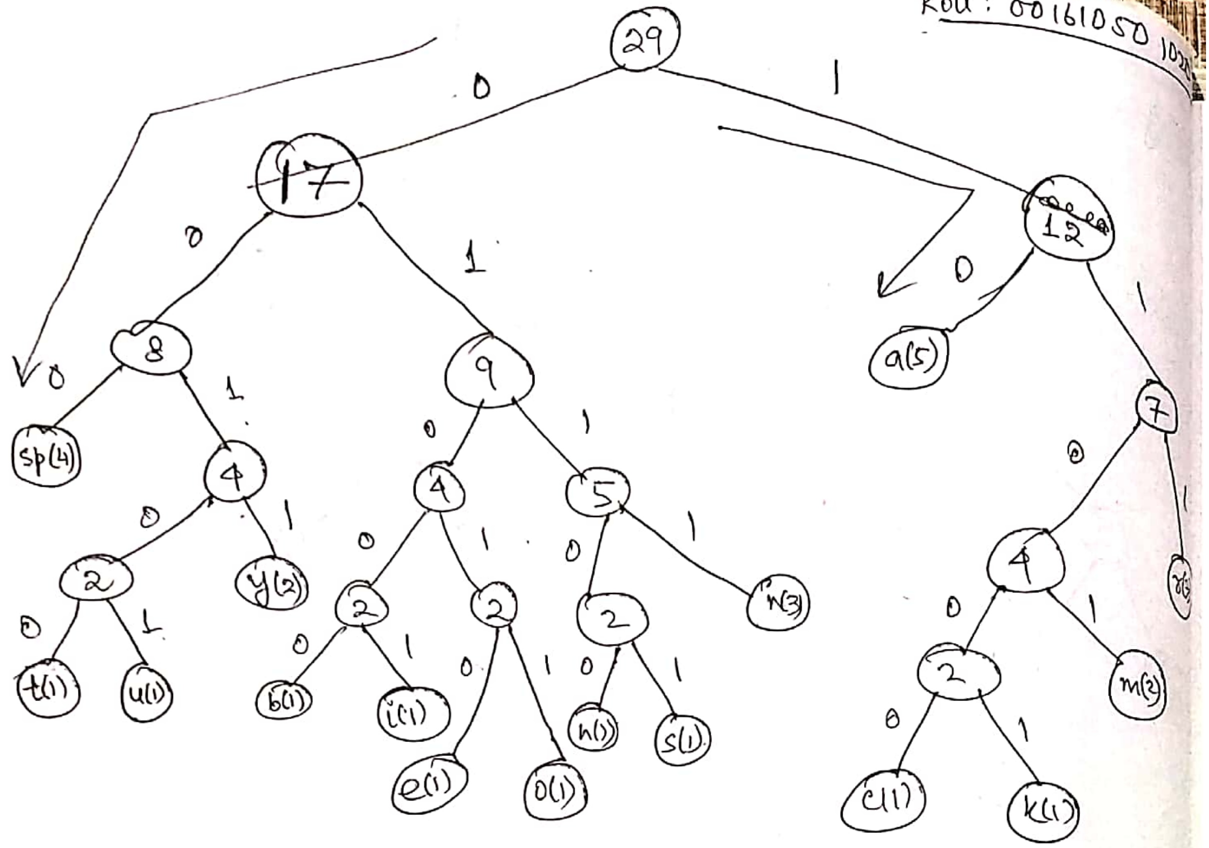






~~Merge b,i,e,~~- Merge remaining and we get final tree.

Roll : 00161050 102



Huffman Codes

~~Ale~~ To find the codes traverse the tree to the character. Left node edge is 0 else 1.

a → 10
space – 000.
n – 0111
r – 111
m – 1101
y – 0011
e – 01010.
i – 01001
s – 01101
u – 00101
c – 11000.
~~h~~ – 01100.
k – 11001

b – 01000
o – 01011
t – 00100.

Anuran Chakraborty

Scanned with CamScanner

Roll : 001610501020

Now concatenate the codes to get the encoded message

my name is anuran chakraborty

1101	0011	000	0111	10	1101	01010	000	01001	01101	000
m	y	sp	n	a	m	e	sp	i	s	sp

10	0111	00101	111	10	0111	000	11000	01100	10	11001	111	10
a	n	u	r	a	n	sp	c	h	a	k	r	a

01000	01011	111	00100	0011 .
b	o	r	t	y

In Huffman coding two codes of different length never have common prefix. So in decoding the above code.

11010011000 0111 . - - -

we see that the minimum length string having prefix 1101 is 4. and also no other string has the same code.

1101 → m

0011 → y.

000 → space.

and so on.

whenever we get a ~~code~~ substring we check whether it is a valid code. If it is a valid code then we replace by that character as it will never be a prefix of a larger length code.

Anuran Chakraborty

Scanned with CamScanner