NAME: ANURAN CHAKRABORTY ROLL: 001610501026
EMAIL: ch.anuraan@gmail.com
MOB. NO.: 8902689001.

Image Processing

Q1. (a) A pixel p at coordinates (x, y) has 4 horizontal and vertical neighbours whose coordinates are given by.

$(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$

This set of pixels, called the 4-neighbours of p, is denoted by $N_4(p)$.

The 4 diagonal neighbours of p have coordinates.

$(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$.

and are denoted by $N_D(p)$. These points together with the 4-neighbours are called the 8-neighbours of p denoted by $N_8(p)$.

- 4-adjacency: Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.
- 8-adjacency: Two pixels p and q with values from V are 8-adjacent if q is in $N_8(p)$.
- m-adjacency (mixed adjacency): Two pixels p and q with values from V are m-adjacent if

(i) q is in $N_4(p)$, or

(ii) q is in $N_D(p)$, and $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

where V is the set of values used to define adjacency.

Anuran Chakraborty

Roll: 001610501020

Q1 (b) In intensity slicing we first consider an images as a 3D function mapping spatial coordinates to intensities (height). Now, we consider placing planes at certain levels parallel to the coordinate plane. If a value is on one side of a plane it is rendered in one colour, and if on the other side it is rendered in a different colour.

In general intensity slicing is.

- Let $[0, L-1]$ represent the grey scale.
- Let 0 represent black $[f(x,y)=0]$ and l_{L-1} represent white $[f(x,y)=L-1]$.
- Suppose P planes perpendicular to intensity axis are defined as levels $l_1, l_2, \ldots, l_p$.
- Assuming that $0 < P < L-1$ then the P planes partition the greyscale into P+1 intervals $V_1, V_2, \ldots, V_{P+1}$.
- Grey level color assignments can then be made according to the relation

$$f(x,y) = c_k \quad \text{if } f(x,y) \in V_k.$$

c_k is the colour associated with the kth intensity level V_k.

Anuran Chakraborty

Roll: 001610501020

Q1 (c) Adaptive filtering means changing the behaviour according to the values of the grayscales under the mask.

$$m_f^2 + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_g^2}(g - m_f)$$

m_f = mean under the mask.
σ_f^2 = variance under the mask.
σ_g^2 = variance of the image.
g = current grayscale.

If σ_f^2 is high fraction is close to 1 and the output is close to g. This is the case case for significant detail such as edges.
If variance is low then output is close to m_f.

Adaptive median filtering is used to remove salt and paper noise, etc.
The median filter performs relatively well on impulse noise as long as the spatial density of the noise is not large.
The adaptive median filter can perform better. The filter size changes depending on the characteristics of the image.

Let, z_{min} = minimum grey level in S_{xy}.
z_{max} = maximum grey level in S_{xy}.
z_{med} = median of grey levels in S_{xy}.
z_{xy} = grey level at coordinates (x, y),
S_{max} = max. allowed size of S_{xy}.
S_{xy} is the window size at (x, y)

Anwaan Chakraborty.

Level A: $A1 = Z_{med} - Z_{min}$.
$A2 = Z_{med} - Z_{max}$.

if $A1 > 0$ and $A2 < 0$, Goto level B.
else increase window size. If window size $\leq S_{max}$ repeat level A. else output Z_{med}.

Level B: $B1 = Z_{xy} - Z_{min}$.
$B2 = Z_{xy} - Z_{max}$

if $B1 > 0$ and $B2 < 0$, output Z_{xy} else output Z_{med}.

Q1(d) chain code is a lossless compression algorithm for monochrome images. The basic principle of chain codes is to separately encode each component, or "blob" in the image. For each such region, a point on the boundary is selected and its coordinates are transmitted. The encoder then moves along the boundary of the region and, at each step, transmits a symbol representing the direction of the movement. This continues until the encoder returns to the starting position, at which point the blob has been completely described, and encoding continues with the next blob in the image. This encoding method is particularly effective for images consisting of a reasonably small number of connected components.

Anuran Chakraborty

Q1(e) Unsharp masking is used to sharpen images. It consists of subtracting a blurred version of an image from the image itself. This process called unsharp masking is expressed as.

$$f_s(x,y) = f(x,y) - \bar{f}(x,y).$$

where $f_s(x,y)$ = sharpened image obtained.

$\bar{f}(x,y)$ = blurred version of $f(x,y)$.

The origin of unsharp masking is in dark room photography. However, the resulting image although clearer may be a less accurate representation of the image's subject.

Anwaan Chakraborty

Q2(a) Algorithm for histogram equalization:

Histogram equalisation is spreading out the frequencies in an image in order to improve dark or washed out images.

In order to arrive at a suitable transformation we have to first apply it to continuous function. Let r be the image whose values are normalized between 0 and 1., r=0 is black and r=1 is white. Later we consider a discrete formula and allow pixel values to be in the interval [0, L-1]. we have to find a transformation

$$s = T(r) \quad 0 \leq r \leq 1.$$

that produce a level s for every pixel value r in the original image. Assume,

(a) T(r) is single valued and monotonically increasing in $0 \leq r \leq 1$.

(b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

T(r) must be single valued for inverse to be possible, and monotonicity condition preserves the increasing order from black to white in the output image.

$$r = T^{-1}(s) \quad 1 \leq s \leq 0.$$

The gray levels in an image are random variables $p_r(r)$ and $p_s(s)$ denote probability density functions of r and s. We know, from probability.

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|$$

In IP

$$s = T(r) = \int_0^r p_r(w)\, dw$$

Anwaan Chakraborty

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] = p_r(r)$$

$$\therefore p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\cdot\left|\frac{1}{p_r(r)}\right| = 1 \qquad 0 \le s \le 1$$

$p_s(s)$ is a uniform pdf.

For discrete values r_k

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, \ldots, L-1$$

n_k = no. of pixels having gray level r_k

r_k = gray level.

n = total ~~no of pixels in the image~~ sum of frequency of all levels.

$$\therefore s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

Thus to perform histogram equalisation apply the transformation.

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}$$

to every pixel in the image.

Anuran Chakraborty

Algorithm:

1. Convert input image into grayscale image
2. Find frequency of ~~occurence~~ occurrence for each pixel value i.e. histogram.
3. Calculate cumulative frequency of all pixel values.
4. Divide the cumulative frequencies by total number of pixels and multiply them by maximum graycount in the image.

Anwaan Chakraborty

Roll : 001610501020

Q2(b) Mexicah Hat filter for edge detection :

Mexicah hat filter for edge detection is also known as Laplacian of Gaussian or the LoG filter. It is a high pass filter.

We know the laplacian filter of a 2D function $f(x,y)$ is a second order derivative.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

we know $\frac{\partial f}{\partial x} = f(x+1) - f(x)$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$$

This can be represented by a filter matrix.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

However, the derivative operator such as laplacian is prone to noise so before applying the laplacian filter a Gaussian filter is applied on the image to remove high frequency noise. Also applying only the laplacian can result in double edges. Also the laplacian is unable to detect edge direction. For these reasons we apply LoG.

Anwan Chakraborty

~~The Laplacian~~

The Laplacian is combined with smoothing as a precursor to finding edges via zero crossings.

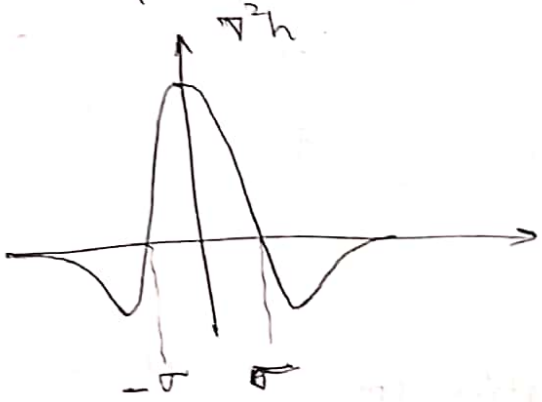
$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

where $r^2 = x^2 + y^2$ σ = std. deviation.

Convolving this function with an image blurs the image, degree of blurring being determined by σ.

$$\nabla^2 h = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

This function is known as the Laplacian of Gaussian (LoG) function.



An approximate 5×5 filter is shown below which captures the shape of the graph

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Sum of the values in the mask must be equal to 0 so as not to change in areas of constant gray level.

Thus the purpose of the Gaussian function is to smooth the image, and the Laplacian is to provide an image with 0 crossings used to establish the location of the edges.

Anuran Chakraborty

Roll: 001610501020

Q2(c) In order to define opening and closing morphological operations we first have to define Erosion and Dilation.

Dilation:

With A and B sets in Z^2, the dilation of A by B, denoted by $A \oplus B$ is defined as

~~$A \oplus B = \{ z \mid (\hat{B})_z$~~

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \phi \}.$$

This is based on obtaining the reflection of B about its origin and shifting this reflection by z. The dilation of A by B then is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element.

Erosion:

For set A and B in Z^2 the erosion of A by B denoted by $A \ominus B$ is defined as.

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}.$$

That is the set of all points z such that B, translated by z is contained in A.

Opening: Opening generally smoothes the contour of an object, ~~to~~ breaks narrow isthmuses, and eliminates thin protrusions. The opening set of A by structural element B may be defined as.

$$A \circ B = (A \ominus B) \oplus B$$

Anuran Chakraborty

Closing: Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour. The closing set A by structuring element B, denoted by $A \bullet B$ is

$$A \bullet B = (A \oplus B) \ominus B$$

Boundary Extraction:

The boundary of a set A, denoted by $\beta(A)$ can be obtained by first eroding A by B and then performing the set difference between A and its erosion.

$$\beta(A) = A - (A \ominus B)$$

Roll: 001610501020

Q3 (a) Weiner filter or minimum mean squared error filter is used for image restoration.

~~Let $\hat{f}(u,v)$ denote the undegraded image and~~

In this method we aim to find an ~~its~~ estimate $\hat{f}$ of the uncorrupted ~~image~~ image f such that the mean squared error between them is minimized. This error measure is given by

$$e^2 = E\{(f - \hat{f})^2\}$$

where $E\{\}$ is the expected value of the argument. Generally, we take sum of squared errors. It is assumed that the noise and the image are uncorrelated; that one or the other has zero mean; and that the gray levels in the estimate are a linear function of the levels in the degraded image. Based on these conditions the minimum of error ~~of the~~ function is given in the frequency domain by the expression.

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)\, S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right] G(u,v)$$

$$= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v)$$

$$= \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v)$$

Anwran Chakraborty

where, $H(u,v)$ = degradation function.

$H^*(u,v)$ = complex conjugate of $H(u,v)$

$|H(u,v)|^2 = H^*(u,v)\,H(u,v)$

$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise

$N(u,v)$ = noise function.

$S_f(u,v) = |F(u,v)|^2$ = power spectrum of undegraded image.

$G(u,v)$ = transform of the degraded image.

$\hat{F}(u,v)$ = fourier transform of the undegraded image.

When dealing with spectrally white noise, the spectrum $|N(u,v)|^2$ is constant.

If $|F(u,v)|^2$ is not known

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

K is a constant.

Anuran Chakraborty

Roll : 00161050102O

Q3(b). The derivatives of digital functions are defined in terms of differences. There are many ways to define these differences. We require that any definition we use for a first derivate:

(1) must be zero in flat segments (areas of constant gray-level values)

(2) must be non-zero at the onset of a gray level step or ramp.

(3) must be non-zero along ramps.

The basic definition of the first-order derivative of a one-dimensional function $f(x)$ is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial f}{\partial y} = f(y+1) - f(y)$$

$$\nabla f(x,y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

$$= f(x+1,y) - f(x,y) + f(x,y+1) - f(x,y)$$

$$= f(x+1,y) + f(x,y+1) - 2f(x,y)$$

$\therefore$ filter $= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Anuran Chakraborty

Roll : 001610501020

First derivatives in image processing are implemented using the magnitude of the gradient. For a function $f(x,y)$, the gradient of f at coordinates (x,y) is defined as the two dimensional column vector.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of this vector

$$\nabla f = \text{mag}(\nabla f)$$

$$= \left[G_x^2 + G_y^2\right]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \quad \text{—(1)}$$

Since the above equation is computationally expensive and not trivial we approximate the magnitude by using absolute values.

$$\nabla f \approx |G_x| + |G_y| \quad \text{—(2)}$$

Now let a window by represented by

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

z_6 denotes $f(x, y)$.

z_1 denotes $f(x-1, y-1)$

Simplest approximation to a first order derivate is $G_x = (z_5 - z_8)$ $G_y = (z_6 - z_5)$

if we use the exact calculation of eq: (1)

$$\nabla f = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}$$

Anuran Chakraborty

If we use eq^{n} (2)

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Also if we use Robert's cross differences

$$G_x = (z_9 - z_5) \qquad G_y = (z_8 - z_6)$$

then using eq^{n} (1)

$$\nabla f = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

using eq^{n} (2)

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Another approximation using 3×3 filter mask is

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

This may be represented by the masks

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(a) (b)

The above masks (a) approximates the derivative in x-direction. (b) approximates in y-direction. The above masks are called Sobel operators. The idea behind using a weight of 2 is to achieve smoothing by giving more importance to the center point. The mask coefficients sum to 0 thus gives 0 in area of constant gray level as expected.

Anuran Chakraborty

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

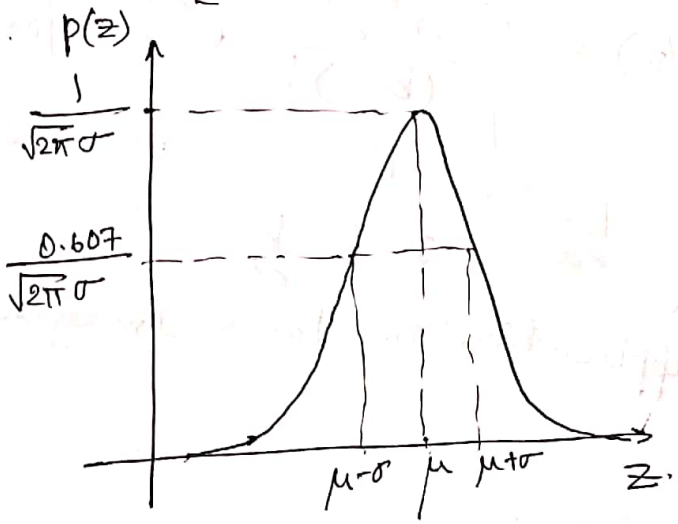
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

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 $[(\mu-2\sigma), (\mu+2\sigma)]$.



Anwran Chakraborty

• 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

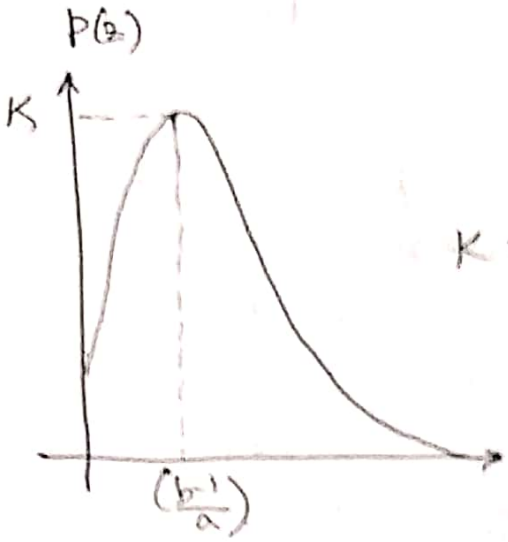
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq a. \\ 0 & \text{for } z < a. \end{cases}$$

$$\mu = b/a \qquad \sigma^2 = \frac{b}{a^2}$$

$a > 0$, b = +ve integer

It has application in Laser Imaging.

Anwesan Chakraborty



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

Q4(b). Harmonic Mean filter.

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

It is given by the equation

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

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

$g(x,y)$ = pixel value of 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.

Anwaan Chakraborty

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

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.

Roll: 00161050102O

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

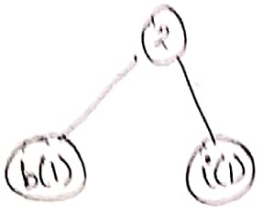
a(5) sp(4) n(3) r(3) m(2) y(2) e(1) i(1) s(1)

Anuran Chakraborty

Roll: 001610501020.

1. Select highest priority characters.

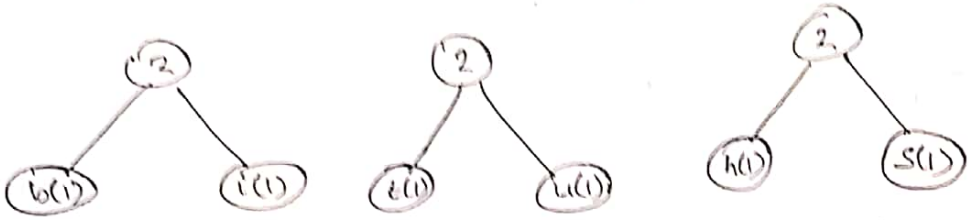
choose b, i



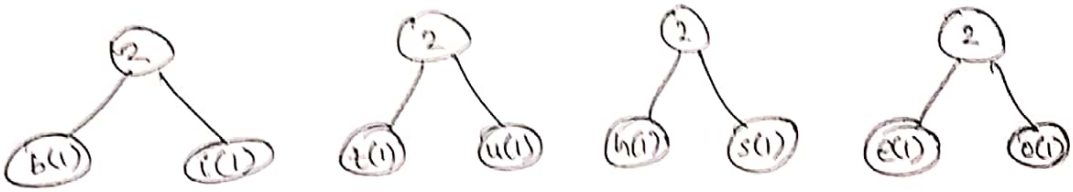
Remaining

choose t, u.

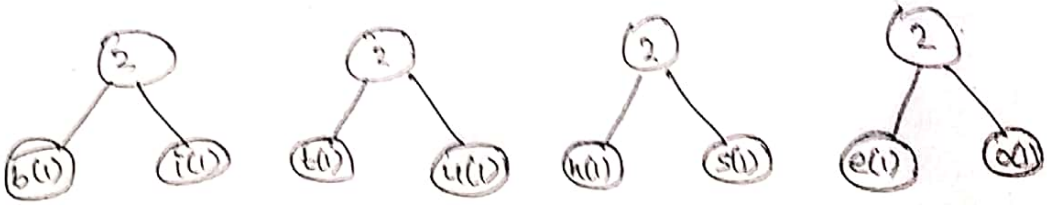
choose h, s.

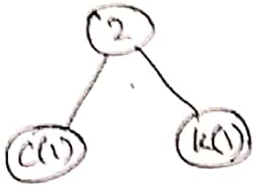


choose e, o.



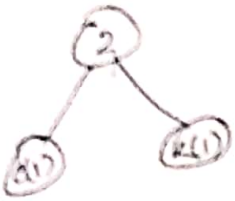
choose c, k



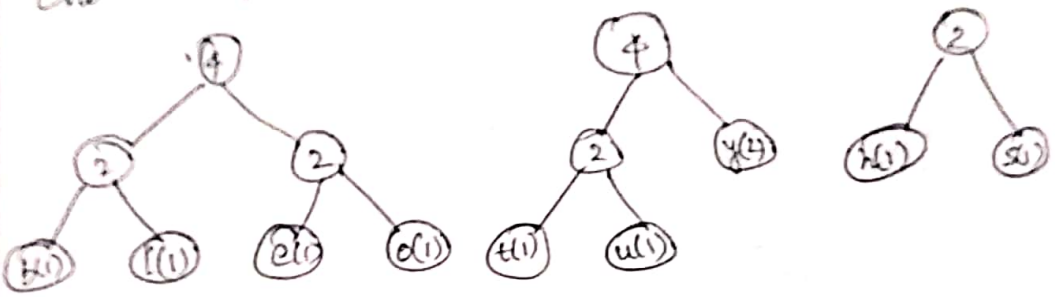


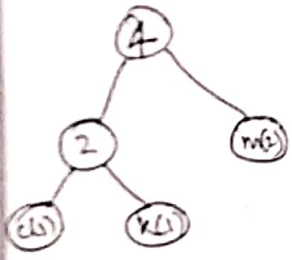
Anuran Chakraborty

choose t, u and y as both have freq 2

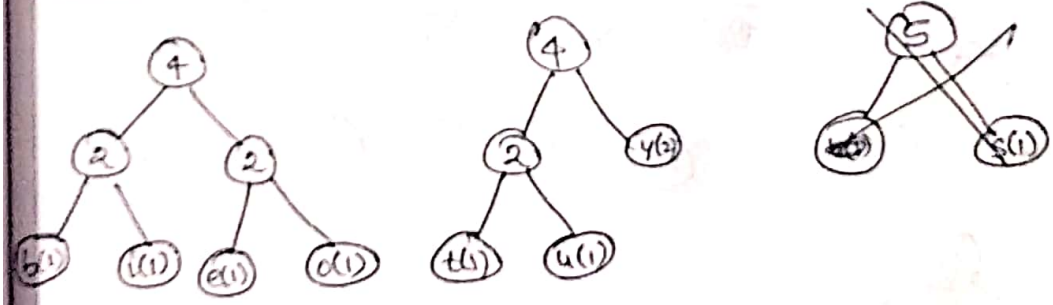


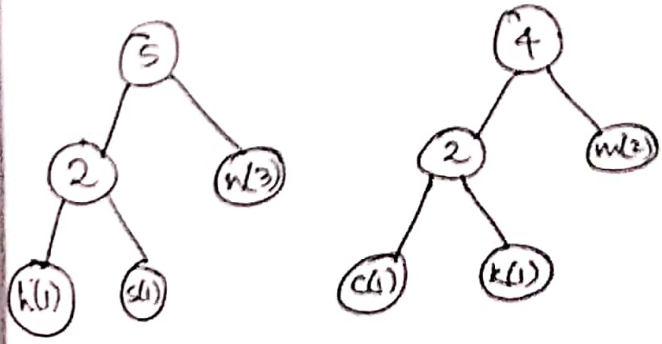
choose b, i and e, o | and c, k and m





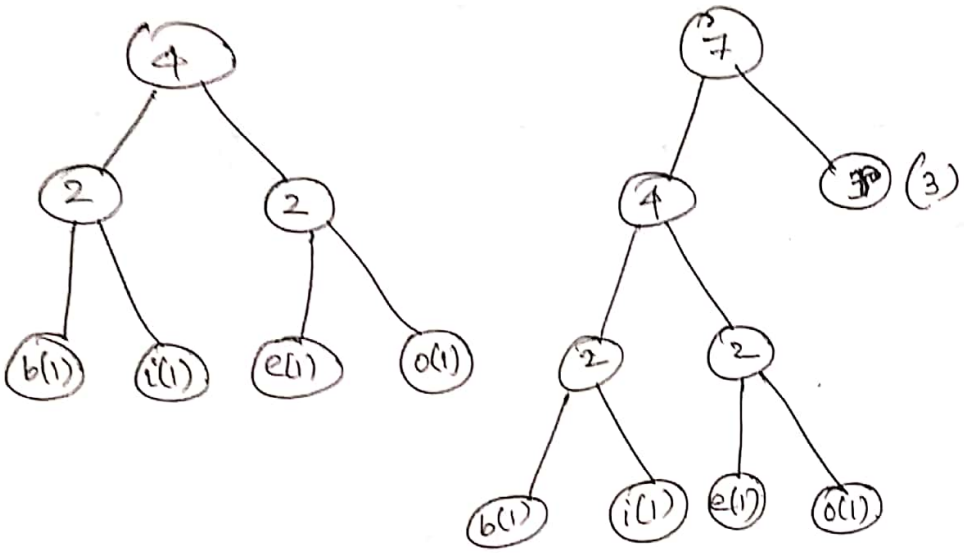
Choose h, s and n.

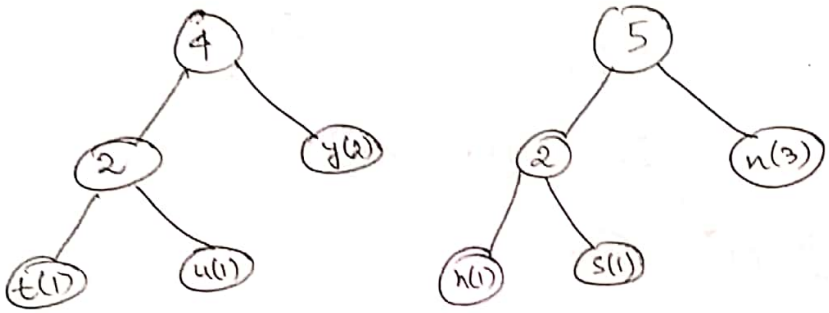




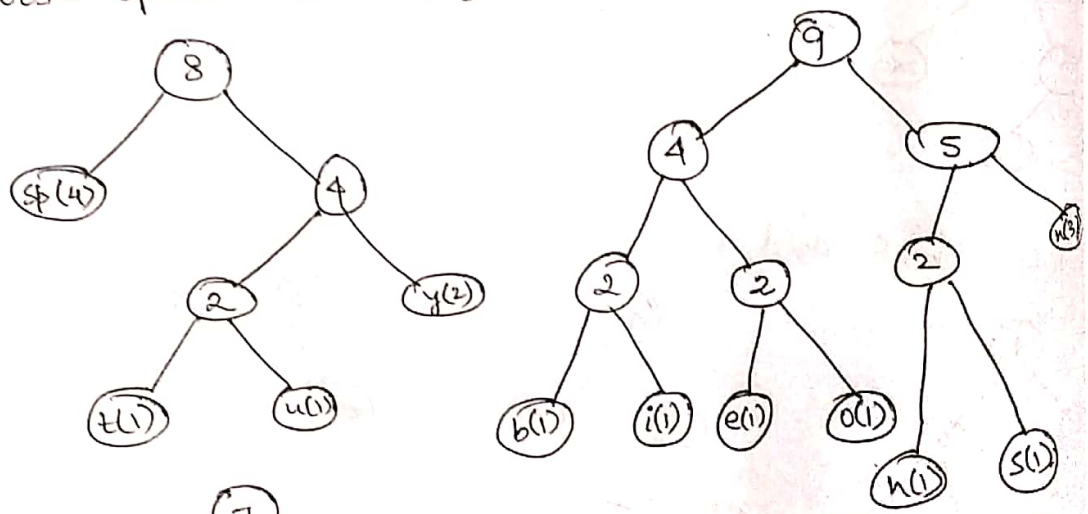
Anuran Chakraborty

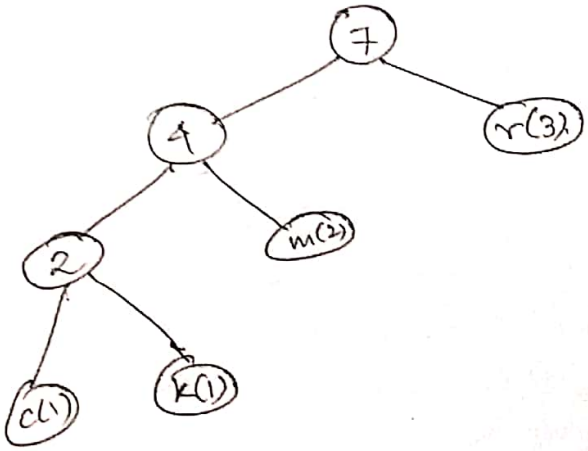
Next choose c, k, m and r.





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

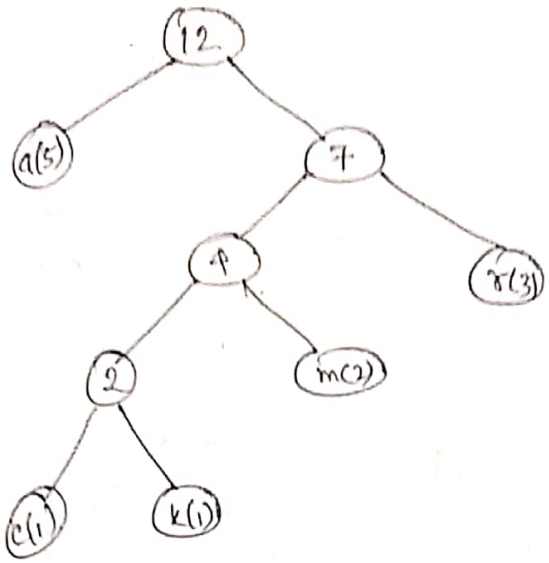


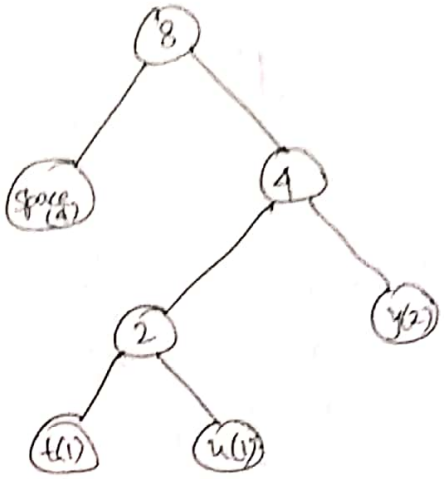


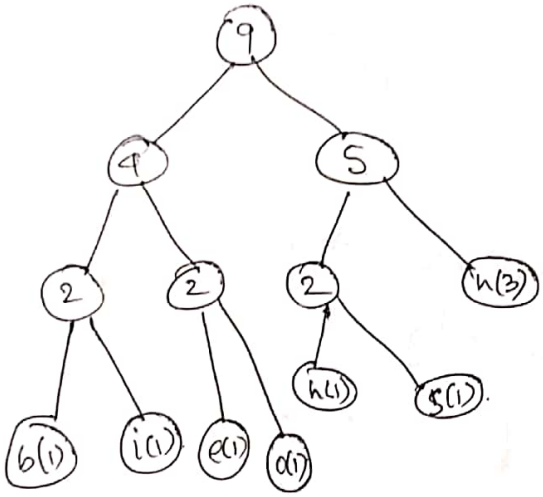
Anuran Chakraborty

Roll : 001610501020

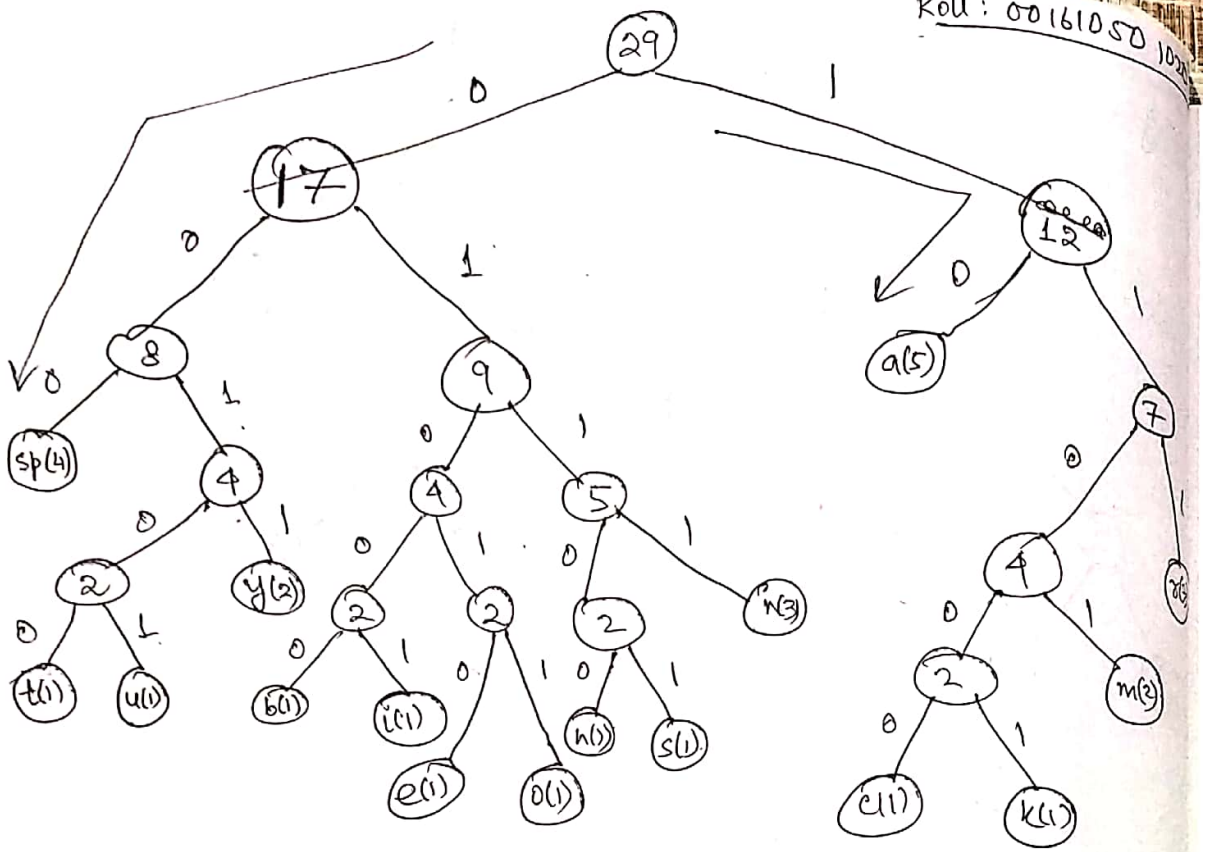
Choose ~~Merge~~ a(5) with c, k, m, r.







~~Merge twice~~,- Merge remaining and we get final tree.



Huffman Codes

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

Anwesan CRakraborty

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 11100100 0011
b o r t y .

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

1101 0011 0000 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 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

Suppose the above tree is given and the string my is encoded as.

11010011

We traverse the string.

1. Get 1. Go right from root.
2. Get 1. Go right
3. Get 0. Go left.
4. Get 1. Go right. This is a leaf node with alphabet m. Thus str = m.
5. Go back to root.
6. Get 0. Go left.
7. Get 0. Go left.
8. Get 1. Go right
9. Get 1. Go right. This is a leaf node with alphabet y. Thus str = my.

This continues for the whole encoded string.

Anuran Chakraborty