

DNA Origami

Folding of DNA strands to get patterns.

Tools :

- CADNano.

Difficulties due to folding of base pairs etc (as compared to paper).

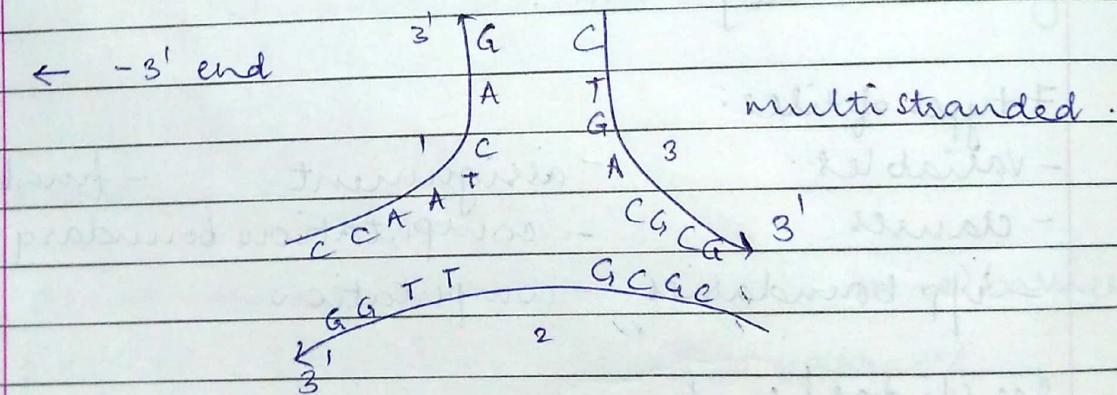
"art of a nerd".

Applications are still in testing phase.

Self assembly as one of the ways to create shapes → like crystalline molecules.

A B C D → Components.

Design glues for each of them to come together.
In DNA ; the glues will not be necessary due to complements. (Watson Crick model).



(first half of one, second half of two) → complements

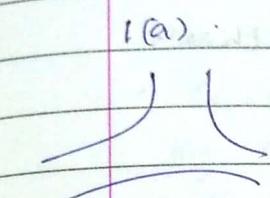
Another eg: DNA like a plait or as hair is used in a braid.

Some strands participate in two helices:

(see slides)

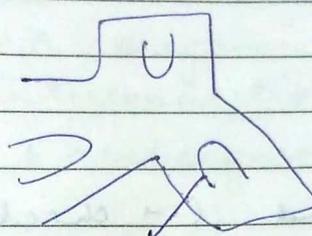
Approaches to create a design :

1. designs classified by how they are built for component strands.
- a) Composed of entirely of short oligonucleotides.
- b) " " one long scaffold strands and many help strands.
- c) one long strand, few helper.

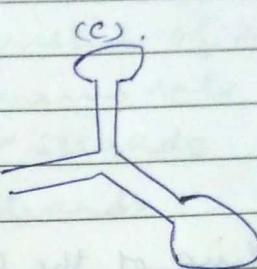


multi stranded.

(b.)



~~need keep~~
main strand in
place.



complements ~~base~~ b/w
the strand.

↓
only t.

scaffold origami

single stranded.

Based on strands repeated:

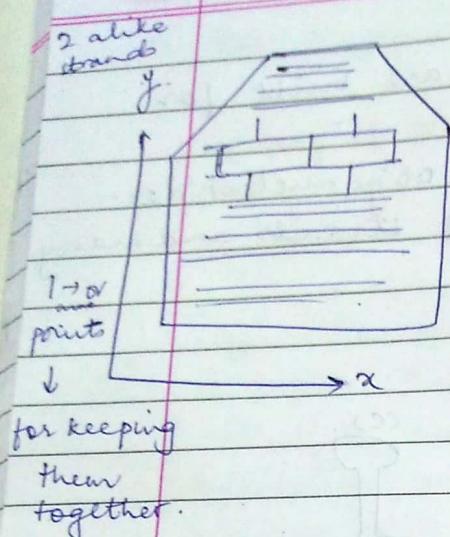
- 1) Unique strands → would be a one pot reaction.
- 2) multiple repeated strands.

* Design of scaffolded DNA origami.
DNA : 2 nm in diameter.

Every turn : 3.6 nm of helical strand.

Steps :

- 1) Build a geometric model of a DNA structure that approximates the desired shape.



cylinder filled up
→ 1 cylinder

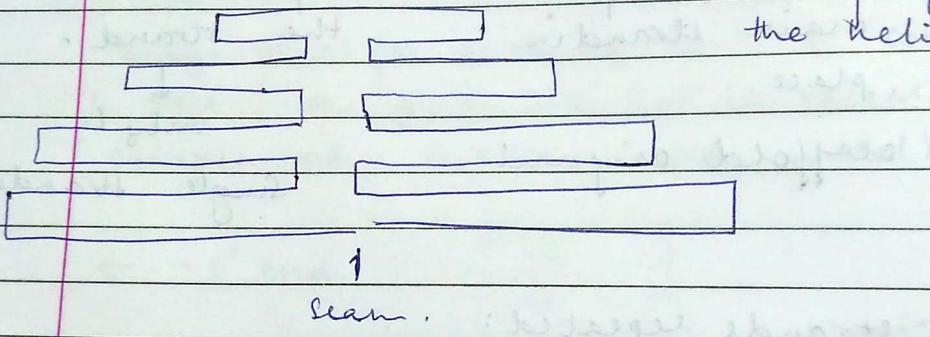
rule for length : integral number of turns.

length should be a multiple of 3.2.

y dirⁿ : 2 strands together.

2) Folding of the strand (→ check slides).

Folding of the strands on the helix.



3) Design a set of staple strands.

Computer determines which ones are the staple strands to make crossover.

DC C → helper strands added to strands from the previous step.

4) twist of scaffold crossovers is calculated and their position is calculated and position is changed.

Crossover → strands getting shared over helixes.

5) In order to achieve higher binding specificity, and higher binding energy which results in higher melting temp pairs of adjacent staples are merged.

Design of 3D origami:

- Bundle neighbouring DNA ~~helices~~ by crossones acc. to the structural characteristics of DNA.
- Join 2D DNA origami domains into 3D layouts by complex interconnection of strands. DNA box → by joining 6 rectangular strands.

Applications :

Drug delivery.

Visualisation of enzymatic sites.

see slides.

Quantum Computing

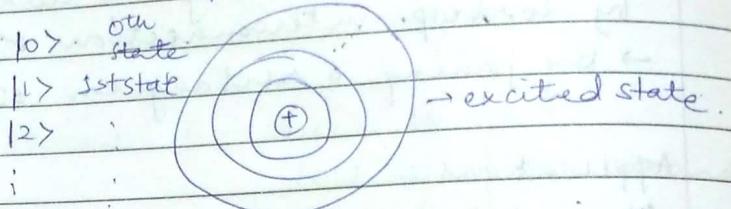
Computers \rightarrow bits \rightarrow abstract concept (no implementation)
 In classical computers we operate on bits : 0 or 1.

How do we implement? ON or OFF state

Quantum main bit is qubit. (quantum bit).

Motivation?

\downarrow
 inspired from E. in atoms.



$|0\rangle$ 0th state
 $|1\rangle$ 1st state
 $|2\rangle$:
 \vdots
 $|k\rangle$ so my qubit can be in any of the k states.

But given an e, you cannot exactly determine which state it's in.

so superposition of states.

State of e :

$$\alpha_1 |0\rangle + \alpha_2 |1\rangle + \alpha_3 |2\rangle + \dots + \alpha_{k+1} |k\rangle.$$

\downarrow
 α belongs to complex space.

$$|\alpha_1|^2 + |\alpha_2|^2 + \dots + |\alpha_{k+1}|^2 = 1.$$

or

Start
from
fund.
will be
done.

when we go to measure the state, we can only see the probabilities $\rightarrow \alpha_i^2$ \rightarrow you lose the state when you go to measure it.

$\alpha_i \rightarrow$ amplitude

So state of an e $|e\rangle$ is a superposition of all states.

single qubit system?

$| \rangle$ Dirac notation
(vector reprⁿ).

So k states is a k dim vector
Date: 20,
Page No.

when $k=2$, so 0 or 1.

$| 0 \rangle$ → ground state

$| 1 \rangle$ → excited state

qubits.

here

bit can take the value $\alpha|0\rangle + \beta|1\rangle$

so can have states in b/w 0 and 1 as well.
and $|\alpha|^2 + |\beta|^2 = 1$.

$$| 0 \rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ k \end{pmatrix}$$

k main
first 1.
base.
 $k-1$
 $k=0$

$$| 1 \rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ k \end{pmatrix}$$

Single qubit system $\rightarrow \alpha|0\rangle + \beta|1\rangle$

if $\alpha=0$

$\beta=1$,

then state is 1

but if $\alpha=\frac{1}{\sqrt{2}}$ and $\beta=\frac{1}{\sqrt{2}}$,

α & β are complex numbers.

then state is $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. \rightarrow this state $\rightarrow \cos \theta / 2$
also called \star for when $\theta = 0$ and $\theta = \pi$.

So single qubit system mein $2^1 = 2$ possible amplitudes.

$$\cos | 0 \rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } | 1 \rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

So, generalized qubit:

$$| \Psi \rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow \text{so 2D vector in a complex space.}$$

$$\text{now } i^2 = -1.$$

$$a+ib$$

↳ its magnitude is $\sqrt{a^2+b^2}$.

So now if we take 2 bit,

possible states are $00, 01, 10, 11 = 2^2 = 4$



* and for each of these we'll have an amplitude.

Bell state or EPR pair

$$|00\rangle + |11\rangle$$

$\sqrt{2}$

so, 2 bit qubit system (under multiple qubit system)
 $\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$.

$\{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\}$ E complex no.s

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1.$$

now measuring a state is impossible in QM.
consider $|0\rangle + \beta|1\rangle$.

so, what's the probability that the state is $|0\rangle$?

$$|\beta|^2$$

the moment I measure this probability, we lose the state and my state becomes $|0\rangle$. \rightarrow collapses to a new state.
see \downarrow coz reduces to

now, measuring 2 bit qubit system.

$$\frac{|0\rangle}{\sqrt{2}}$$

what's p of first qubit being 0?

$$|\alpha_{00}|^2 + |\alpha_{01}|^2$$

\hookrightarrow now state gets reduced to $\alpha_{00}|00\rangle + \alpha_{01}|01\rangle$.

but not valid coz square wont add upto 1.

So, normalize it.

(see why this isn't a problem in single qubit system.)

$$\cancel{\text{Ans}}: \alpha_{00}|00\rangle + \alpha_{01}|01\rangle$$

can only 1 amplitude be 0 at a time?).

$$\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}$$

(so now summation of these new coefficients will be 1)

Gates:

$$\text{Not : } \begin{matrix} 1 \rightarrow 0 \\ 0 \rightarrow 1 \end{matrix}$$

So what should not in qubits do?

$$|0\rangle \rightarrow |1\rangle$$

$$\text{and } |1\rangle \rightarrow |0\rangle$$

but it's a superposition of states. So, not is just these 2 states but more.

So, α becomes β , β becomes α .

$$\text{So, } \alpha|0\rangle + \beta|1\rangle \rightarrow \beta|0\rangle + \alpha|1\rangle$$

Not is a single qubit gate.

(multiple is not...?).

0 to 1, 1 to 0.

What'll the matrix look like?

coz single qubit, so 2 amps, so 2×2

$$\begin{matrix} & & & \\ & & & \\ |0\rangle & 0 & 1 & \\ & & & \\ |1\rangle & 1 & 0 & \end{matrix} \rightarrow \text{So this is NOT gate.}$$

\downarrow so 0

coz you have to make it.

want α

to go to $|1\rangle$

and β to $|0\rangle$

\downarrow 0

So, α should go to 1,

$$\begin{matrix} & & & \\ & & & \\ |0\rangle & 0 & 1 & \\ & & & \\ |1\rangle & 1 & 0 & \end{matrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{matrix} & & & \\ & & & \\ |0\rangle & 0 & 1 & \\ & & & \\ |1\rangle & 1 & 0 & \end{matrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

not gate

vector

vector

not gate

vector

(yaaahh, α were according to

So, generalising,

$$\begin{bmatrix} 0 & 1 \\ \alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

not gate

H.W. \bar{Z} gate \rightarrow single qubit gate

keeps 0 unchanged, flips 1. (makes it -ve).

so if $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle - \beta|1\rangle$.

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

\downarrow \hookrightarrow 1 vector negative (negated)
vector
as it is

can any matrix be designed which gives a certain op be a gate?

No.

for any gate U , $UU^+ = I$. Then it'll be called unitary matrix.

U^+ is transpose and complex conjugating it.



$$\text{So, } U = \begin{vmatrix} a+bi & a+di \\ a+ci & a+ei \end{vmatrix}$$

\downarrow transpose

$$\begin{vmatrix} a+bi & a+ci \\ a+di & a+ei \end{vmatrix}$$

Complex conjugate.

$$U^+ = \begin{vmatrix} a-bi & a-i \\ a-di & a-ei \end{vmatrix}$$



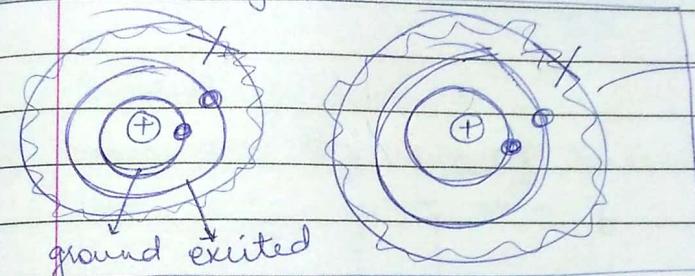
Tensor product

$|00\rangle$
is equiv. to

$|0\rangle \otimes |0\rangle$

is equiv. to
 $|0\rangle|0\rangle$

basically,



ye last waala nahi
only 2 orbitals/states
ground & excited

this entire system will give $|00\rangle$ or $|01\rangle$ or
 $|10\rangle$ or $\cancel{|11\rangle}$.

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha_1|0\rangle + \beta_1|1\rangle)$$

↓
tensor product

$$\alpha\alpha_1|00\rangle + \alpha\beta_1|01\rangle + \beta\alpha_1|10\rangle + \beta\beta_1|11\rangle.$$

so basically $|00\rangle$ is $|0\rangle \otimes |0\rangle$.

In two qubit systems :



measuring first qubit can give off $0 \downarrow 0 \downarrow$.

$$P = \frac{1}{2} \quad P = \frac{1}{2}$$

now, if it were

try measuring

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \beta_0|10\rangle + \beta_1|11\rangle$$

perform a measurement on the first qubit.

by visualizing the vectors in the quadrants

Date :
Page No.:

first outcome will be 0

either 0

or 1

for 0: $|00\rangle^2 + |01\rangle^2 \rightarrow$ state is $\alpha|00\rangle + \beta|01\rangle$

$$\sqrt{|\alpha|^2 + |\beta|^2}$$

$$1: |\chi_{10}\rangle^2 + |\chi_{11}\rangle^2$$

Design a Hadamard gate / square root of not gate

$$0 \rightarrow \frac{0+1}{\sqrt{2}}$$

$$1 \rightarrow \frac{0-1}{\sqrt{2}}$$

$$Z = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

see slides.

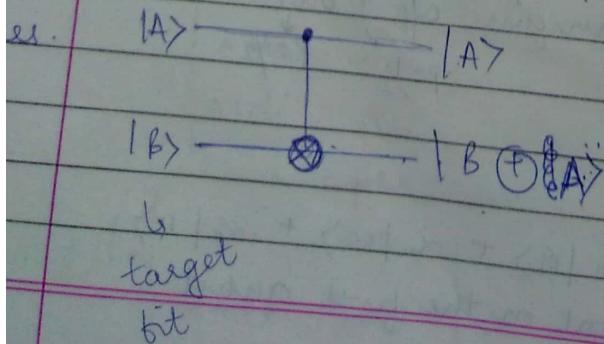
Now, multiple qubit gates :

4x4.

↓

each column depicting 00 01 10 11 → order important.

→ control bit controlled NOT (C-NOT)



$$\text{So, } 00 \rightarrow 00$$

$$01 \rightarrow 01$$

$$10 \rightarrow 11$$

$$11 \rightarrow 10$$

If CB=0 TB unchanged
flipped.

So for CNOT :

	00	01	10	11	
00	1	0	0	0	α
01	0	1	0	0	β
10	0	0	0	1	γ
11	0	0	1	0.	δ

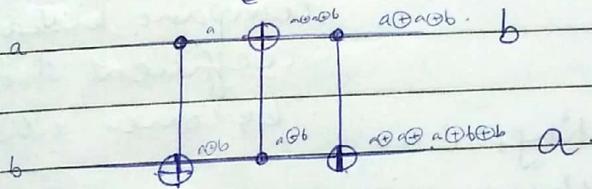
measurement operation.



converts qubit to classical.

Quantum Circuit .

Q write the op of this circuit for ip $|a, b\rangle$



so op is $|b, a\rangle$.

→ see how to write as a tensor product

difference b/w

Quantum circuit

no feedback

no fan in / fan out

Classical circuit

feedback ✓

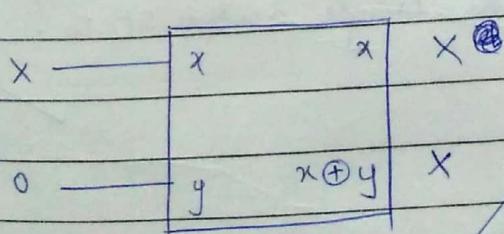
fan in / fan out ✓

Qubit copying state :

can be 0 or 1

X
now we need 2 copies of this.

for quantum,



copying isn't

allowed in quantum

computing coz fan in isn't allowed.

$$|\Psi\rangle = a|0\rangle + b|1\rangle$$

$$|0\rangle \oplus \begin{cases} a|00\rangle \\ b|11\rangle \end{cases}$$

next page

here we're giving $|y\rangle$ as tensor product and hoping it gets copied.

$$\begin{array}{c} \alpha|00\rangle + \beta|11\rangle \\ \text{---} \\ \otimes \quad |0\rangle \quad \text{CNOT} \quad |00\rangle + \beta|11\rangle \end{array}$$

but actually, for $|y\rangle = \alpha|00\rangle + \beta|10\rangle$.

we want off: $\alpha^2|00\rangle + \alpha\beta|01\rangle + \beta\alpha|10\rangle + \beta^2|11\rangle$.

but actual off:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \alpha|00\rangle + \beta|11\rangle \quad \text{see CNOT matrix and compare kiska coefficient should become what.}$$

now what's contradicting,

$\alpha=0, \beta=0$ matching with expected.

when

$\alpha=1, \beta=1$ actual and expected matching

but if $\alpha=\frac{1}{\sqrt{2}}$ or something $\beta=\frac{1}{\sqrt{2}}$; we don't see actual and expected ops matching.

∴ you cannot copy qubit states.

doesn't hold for all conditions of α and β .

what happens if I give a multiple qubit state to a Hadamard gate which is single qubit state?

or EPR pairs (P of each of the states)

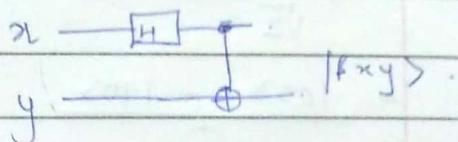
Bell States: (opps basis) is the same)

Hadamard gates operate on the first qubit. (Coz we don't have farin).

So if you give

circuit to generate Bell states.

$$|10\rangle$$



Hadamard operates on 1st bit.

$$\text{So, } \frac{0+1}{\sqrt{2}}$$

then y goes as it is and y ka tensor product.

\downarrow
ipp to CNOT.

first

\rightarrow P.

$$|100\rangle = |10\rangle \otimes |10\rangle.$$

Opp after H,

$$|10\rangle = \frac{|10\rangle + |11\rangle}{\sqrt{2}}$$

ipp to CNOT:

$$\frac{|10\rangle + |11\rangle}{\sqrt{2}} \otimes |10\rangle.$$

$$= \frac{|100\rangle + |110\rangle}{\sqrt{2}} \rightarrow \text{ipp to CNOT.}$$

Opp of CNOT :

$$\frac{|100\rangle + |111\rangle}{\sqrt{2}}$$

try, for ipp $|110\rangle$

$$|110\rangle = |11\rangle \otimes |10\rangle.$$

so now,
 $\text{op} \oplus H$
 $|11\rangle = \frac{10\rangle - 11\rangle}{\sqrt{2}}$

now, ifp to CNOT:

$$\frac{10\rangle - 11\rangle}{\sqrt{2}} \otimes |0\rangle$$

$$= \frac{|100\rangle - |110\rangle}{\sqrt{2}}$$

now, op of \oplus CNOT:

$$\frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$\therefore |\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

Quantum Turing Machine

see classical TM.



transducer → ~~sor~~ writes op on tape

acceptor → checks whether ~~yes~~ final is yes or no (checks language present or not)

Time complexity



number of steps it take

Space complexity



number of cells you'll be utilising.

Configuration of TM:

Transition matrix:

↳ maps configⁿ against another configⁿ.

↳ state s_n , where head points to, contents of tape [these are configⁿ of ATM].

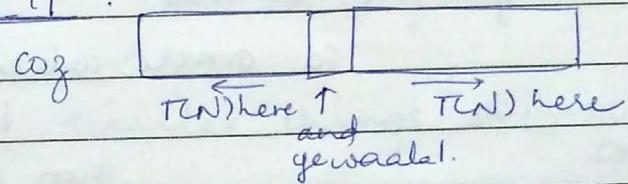
So, in a transition matrix,
 $G \ C_2 \ C_3 \dots \ C_N \rightarrow \text{configs.}$

C_1				
C_2				
C_3				
\vdots				
C_N				

say if $G = (q_1, 1011, \overset{1}{l})$.

state ✓ ✓ position → these 2 are ∞ .
 r_n contents → so how to determine
 a fixed transition matrix?

since length ∞ , use time complexity, at max I can go $T(N)$
~~steps~~ steps on the right or left. So, I can limit
 the length to $2T(N) + 1$.



so, size of the matrix?

first, let Γ be the tape symbol.

contents of tape can be $\Gamma^{2T(N)+1} \rightarrow$ kyunki har mein
 Tee aayega, $2T(N)+1$ is length.

assume length = 3.
 and $\Sigma = \{a, b, c\}$

now, size ^{total} conf's:

$$= \text{present state} \times \text{contents} \times \text{pos}'s$$

$$= |Q| \times \Gamma^{2T(N)+1} \times 2T(N)+1$$

$\begin{array}{c} \downarrow \\ 3 \end{array} \quad \begin{array}{c} \downarrow \\ 3 \end{array} \quad \begin{array}{c} \downarrow \\ 3 \end{array}$
 so $3^3 = 27$.
 $\begin{array}{c} \downarrow \\ a \end{array}$ length.

what's the size of confⁿ matrix? see.

1 in the matrix if transition exists.

	C_1	C_2	C_3
C_1	..	1	..
C_2
C_3

if a transition takes q_1 from C_1 to C_2 ,
 then put 1 otherwise 0

deterministic?

there's another version

deter. ^{TM w/} oracle. → yes or no.

(call the oracle)

whether string present or not → yes no
 ↓

when TM enters q_0 ^(special query) answer will proceed
 from there.

TM ~~goes~~ asks oracle

whether current string is at present position, whether it's present or not.

so oracle writes yes or no
 ↓

then TM goes into a special post-query state.

Probabilistic TM:

↳ going from one config to another is a probability.

2 parameters: to determine acceptance

soundness - if string \notin lang, " " < soundness

completeness - if string \in lang, should be $>$ completeness

\geq completeness → accepted

\leq soundness → rejected

Quantum Turing Machine

config? coeff? see slides

TCM → has amplitudes

should be unitary.

deterministic?

there's another version
deter. TM oracle. → yes or no.

(Paul the Oracle)

whether string present or not → yes no
→ ↴ ↴

when TM enters q_1 ^(special query) from there.

TM ~~goes~~ asks oracle

whether current string is present position τ , whether it's present or not.

so oracle writes yes or no
↓

then TM goes into a special post-query state.

Probabilistic TM:

↳ going from one config to another is a probability.

2 parameters to determine acceptance

soundness - if string \notin lang, " " $<$ soundness

completeness - if string \in lang, should be $>$ completeness

\geq completeness \rightarrow accepted

\leq soundness \rightarrow rejected

Quantum Turing Machine

config? coeff? see slides

+ CTR \rightarrow has amplitudes

should be unitary.

Plant Inspired Algorithms.

symmetry, fractals etc

Plant propagation behaviour - how they spread

• good env

• good soil

so they choose better stuff

so it can be an inspiration for
optimising algos.

light foraging behaviour -

bending towards light

branching algos

purported swarm behaviour -

↓
on purpose similar things swarm flock together.

• Invasive Weed Optimisation (IWO)

how do they grow

↳ survival of the fittest

↳ and you don't specifically have to grow them.

3 key components

- seeding

- seed dispersal

- competition b/w plants

seed dispersals

are stochastic.

every plant produces a certain number of seeds.

what is the number?

$$\frac{N_x \text{ seeds}}{\text{plant } x} = \frac{\text{fitness}_x - \text{colfitness}_{\min}}{\text{colfitness}_{\max} - \text{colfitness}_{\min}} \quad (\text{cot is colony})$$

Seeds ka number \propto fitness.

$$(N_{\max} \text{ seeds} - N_{\min} \text{ seeds}) + \frac{N_{\min} \text{ seeds}}{N_{\max} \text{ seeds}}$$

vector x , n dimensions.

$$x = (x_1, x_2, \dots, x_n)$$

$\underbrace{\hspace{10em}}$

dispersing over vector.

how to choose n value

↓

normal distⁿ.

$$\mu = 0, \sigma = \text{varies for each iter}^n.$$

$$\text{Filter} = \left(\frac{\text{iter}_{\max} - \text{iter}_{\text{curr}}}{\text{iter}_{\max}} \right)^n (\sigma_{\max} - \sigma_{\min}) + \sigma_{\min}$$

this is any n value.

(most frequently used is 2)

σ decreases every iterⁿ.

so, variance also decreases.

now, after dispersal, competition b/w the plants.

at a time, you can have only P_{\max} number
of plants.

Initial population P_{initial}

only plants with greater fitness value will survive.

see algo-slides.

Paddy Field Algo's

5 stages:

sowing

selection $y = f(x)$ fitness of plant

seedling $s_i = q_{\max} \cdot \min(y_i - y_t)$ minimum fitness

$$(y_{\max} - y_t)$$

pollination

$$v_i = e^{\frac{(v_i - v_{\max})}{v_{\max}} - 1}$$

dispersion

$$s_i^{\text{viable}} = v_i s_i$$

again vector x of n dimⁿ representing locⁿ in n -dimⁿ space.

compute $y = f(x)$.

fitness value for plant at location x .

all the seeds are not useful — only pollinated ones are.
So determine pollinated ones for the next iter.

seedling is relative to fitness in surrounding,
and assumed that only q_{\max} seeds can be produced.
as s_i means all seeds aren't useful.

pollination dependent on the neighbouring plants.

say radius a hai. So number of plants in radius a
are neighbours.

v_i → neighbours of plant i

v_{\max} → max. neighbours

so pollination will be highest for a plant having also
neighbours.

disperse only s_i^{viable} seeds.

the iterⁿ stops only if the stuff generated is same as
previous one generated. → so inspiration for clustering.

DNA Computing & Encryption and Decryption

Requirement of DNA Encryption algos.

1. DNA encoding of complete character set
2. Dynamic encoding table generation
3. Unique sequence of for encoding of every character of plaintext to DNA sequence.
4. Robustness of encoding
5. Biological process simulation
6. Dynamicity of encryption process.

+ Encoding algo

- Encryption & Decryption algo

2 encoding tables — receiver & sender
using it will sequences (intron seq. + 1 will generate
encoding seq. 1
and 2 will gen. 2).

tot will have a DNA encoding table.

Tot encryp" raise?

plaintext to DNA seq.



divide into 2 equal halves.

consider plaintext :

intron

encode
using table 1

encode
using table 2

if string is odd length, append an extra bogus character
and then divide it.

DNA Computing & Encryption and Decryption

Requirement of DNA Encryption algos.

1. DNA encoding of complete character set
 2. Dynamic encoding table generation
 3. Unique sequence of for encoding of every character of plaintext to DNA sequence.
 4. Robustness of encoding
 5. Biological process simulation
 6. Dynamicity of encryption process.
- Encoding algo
 - Encryption & Decryption algo

2 encoding tables — receiver & sender
using intron sequences (intron seq. # 1 will generate encoding seq 1
and 2 will gen 2).

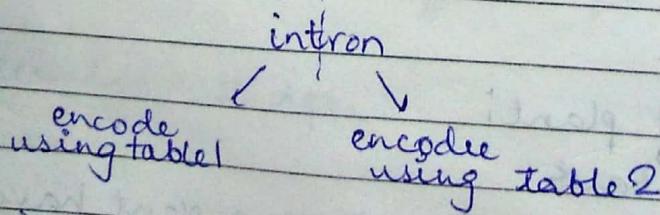
Toh will have a DNA encoding tables.

Toh encryp" kaise?

Plaintext to DNA seqⁿ.

↓
divide into 2 equal halves.

consider plaintext :



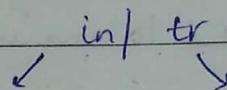
if string is odd length, append an extra bogus character and then divide it.

intron one #

now whatever has been encoded, apply multiple rounds of func's (min 10 rounds).

In every round, then XNOR with introns (encode karne pe binary seqⁿ hain jo, usko introns ke binary seqⁿ se) XNOR hain).

e.g:



has to be XNORed with intron seqⁿ intron seqⁿ se XNOR.

how to convert them to binary?

A - 00

T - 01

C - 10

G - 11

So, 10100011 11100101.

if suppose 10100111 11000011
intron seq: 11111011 11011001

do XNOR

now we need to generate mRNA. (transcription) T → U.
we'll convert back to ATCG form,
then do transcription

↓
don't have Ts,

Uracit in place of Thymine

then mRNA to tRNA (translation)

so just complement it.
(A-U, G-C).

then tRNA to DNA (reverse transcription) U → T.

then right shift the sequences.

TAT C AATT
TTAT CAAT

printing mistakes in paper.
(shifting, decryption etc.).

VISION

Date :
Page No.

but why not directly complement?
↓

coz it should be a simulation of biological processes. Seedha complement nahi.

then after right shift, DNA aayega usko make thkt. then tRNA obtained is converted to amino acids.

↳ using amino acid table.

now the resultant protein sequence is the cipher text.

See amino acid table generation and amino acid gen.

ex:

ATCG , GTAC

randomisation

↓
first convert to mRNA.

AUCA G GUAC.

↓ ↓
cols rows

	A	U	C	G
G	GA	GU	GC	GG
U	UA	UU	UC	UG
A	AA	AU	AC	AG
C	CA	CU	CC	CG

4x4 is extended to 16x16.

So you have 256 possibilities of amino acids.

4! possible ways. → 24

div in 4 groups so 64 each.

4 choose 1

(another level of randomisation)

Ques Given the plaintext "BANK" and the encoding for each letters as

$$B = AAGG$$

$$A = ACAT$$

$$N = GCTT$$

$$K = GAGG$$

} from encoding table 2

The inton / sequence is 0111100001111101.

2

10000110111101

Encrypt the plaintext. Show only single round of multi-round fn.

$$A \rightarrow 00$$

$$T \rightarrow 01$$

AA G₂ G ACAT

GCTT GAGG

$$C \rightarrow 10$$

00 00 11 11 00 10 00 01

11 10 01 01 11 00 11 11

$$G \rightarrow 11$$

01 11 10 00 01 11 11 01

10 00 01 11 01 11 11 01

1000 1000 10 1000 11

0001 11 01 01 00 11 01

C A C A C C A G

C T G T T A G T.

mRNA : CA CA CC AG

CU GU UA GU

tRNA : GU GU GG UC

GA CA AU CA

DNA : GT GT AA TC

GA CA AT CA.

RS : CG TG TG GT

AG AC AA TC

now again,

mRNA : CGUG UAGU

A GAC AAUC

tRNA : GCAC ACCA

UCUG UUAG.

amino acids : MB IB

W.C WI

then, DNA transformed with inton
transformed DNA converted to plaintext and merged.
See decryption process from the slide.

Cellular Automata

you have an initial state.



Cells evolve w/time



patterns based on

transition functions. (change based on transition
fns).

we have a lattice of cells

set of allowable states

transition functions.

will consider square shape.

2 states → alive or dead

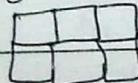


K actually

$k \geq 2$.

$t=1$,

at $t=0$



→ when you apply a trⁿ function, it
changes to one of the k states.

this is one
state.

depending on the neighbourhood.

counting
the cell

not
counting
the cell.

$2r+1$
radius

Features of cellular automata:

- Homogeneity
- Parallelism
- Locality.

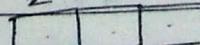
for $r=1$, $k=2$

number of total neighbourhood states
 $(2r+1)^k$

$$2 \times 2 \times 2 \rightarrow k$$

see graphical

depiction in slides.



○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

○ ○ ○

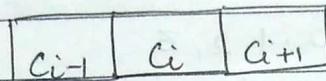
○ ○ ○</p

Transition function

ex: $c_i(t+1) = [c_{i-1}(t) + c_i(t) + c_{i+1}(t)] \bmod 2$.

↑
cell into consideration at time t.

for each of the neighbourhood states, will be having an output.



at t , $c_{i-1}(t)$ $c_i(t)$ $c_{i+1}(t)$.

will the same rule generate the same pattern every time?

↓
depends on time instance
and initial cell value be dependent.

eg: $t=0$

0	1	0
---	---	---

 \rightarrow $1+0+0 \bmod 2$
 $0+0+1 \bmod 2$ $0+1+0 \bmod 2$
 $\frac{1}{2}$ $\frac{1}{2}$

$t=1$	1	1	1
-------	---	---	---

$t=2$	1	1	1
-------	---	---	---

what's the number of rules we can have for $r=1, k=2$

8 possible
neighbourhood states

$$2^8 = 256$$

generalising $K^{(2r+1)}$ ↓ ↓
elementary cellular automata.

each rule string will be different.

So, rule 150 is 10010110.

→ is there any problem with this?

000
each can have 2 possible ops.

First order CA: " " " t.

Second order CA: depends on states t and t-1 See "

$$\therefore 2^{(8)} = 256$$

$$= 2^{(K^{2r+1})}$$

First Order: at $t=1$, dependent on neighbourhood at $t=0$.

Second order: at $t=2$, $u \leftarrow u + t=1 - t=0$.

VISION

Date: _____
Page No. _____

Totalistic rules.

↓

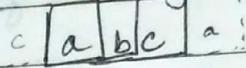
take sum of neighbourhood cells.

find out how many totalistic rules?

↓

status 2 and can be 0, 1, 2, 3
have codes.

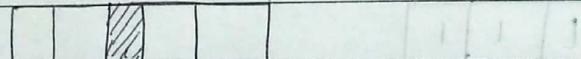
Boundary condition: (whether you wrap...)

- periodic (wrap). 
- reflective
- fixed

Some elementary cellular automata.

Example of rule 0: ($\lambda=1, k=2$)

at $t=0$: a b c d e



neighbourhood = 2 $\lambda+1$.

neighbourhood of b = 0 0 1 (abc).

O/p in rule 0 = 0.

so, b changes to  (0) at $t=1$.

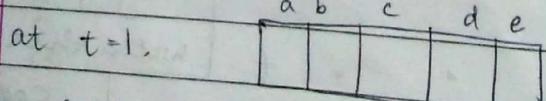
neighbourhood of c = 0 1 0 (bcd).

O/p in rule 0 = 0.

neighbourhood of d = cde 1 0 0

O/p in rule 0 = 0

using wrapping for boundary cells, a and e change to 0.



rule 0 only produces 0.

examples of Rule 1 ($\lambda=1$, $k=2$)

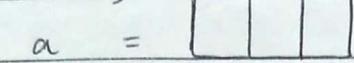
at $t=0$, $a \ b \ c \ d \ e$



.....
.....
.....
.....
.....

example of rule

$t=1$, $a \ b \ c \ d \ e$



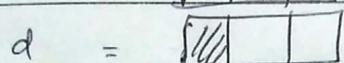
$$O/P = 1$$



$$O/P = 0$$



$$O/P = 0$$



$$O/P = 0$$

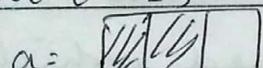


$$O/P = 1$$

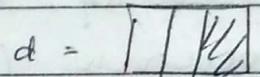
$a \ b \ c \ d \ e$

\therefore at $t=1$, we get

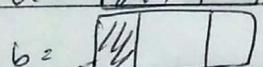
at $t=2$,



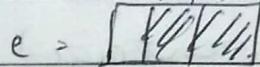
$$O/P = 0$$



$$O/P = 0$$



$$O/P = 0$$



$$O/P = 0$$



$$O/P = 1$$

$a \ b \ c \ d \ e$

\therefore at $t=2$, we get

rule number 1 can be written as :

1 1 1	{	0	}	rule
1 1 0		0		
1 0 1		0		
1 0 0		0		string.
0 1 1		0		
0 1 0		0		
0 0 1		0		
0 0 0		1		

eg: for $a=2$

check

↳ some short

(a) moon neighbourhood (8x8 cells).

(b) Van Neumann neighbourhood (4x5 cells).

Neighbourhood = in 2D CA:

disadvantages)

for decoding, keep the ciphertext with the central cell for decoding, this needs to be stored, which is a 3x3 column.

xor it with the plaintext to get the ciphertext.

column. (it is a random number).

Then, we get a new key. Take its carry

well get our initial nxs config. Now, apply rule 30 until

as bits

value is

and

↑

Hello

0s + 1s

n x 8 array

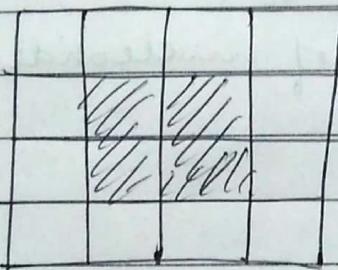
shown by this square test).

Using rule 30 for encryption (leads to random a

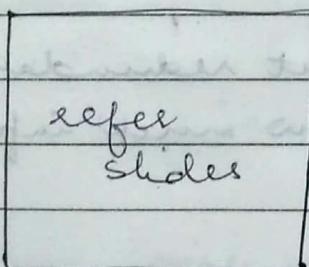
Game of life : Rules

- 8 cells Moore neighbourhood is needed to determine the state of the central cell.
- A dead cell becomes alive at the next step if exactly 3 of its neighbours are alive.
- An alive cell remains alive if 2 or 3 of its neighbours are alive.
 - If only 1 is alive, cell dies of loneliness.
 - If 4, 5, 6, 7, 8 are alive, cell dies of overpopulation.

Game of Life : Configurations



= Invariant



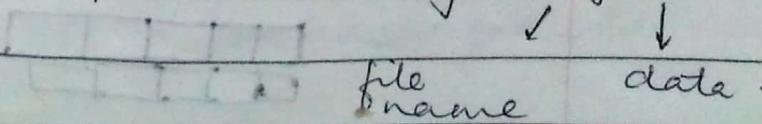
refer sides

blinker

DNA Based Archival Storage System

We went for DNA storage due to space (we wish to store more number of bits per unit area).

Random access is made possible using (key, value) pairs.



Key is a primer and then we apply PCR.

[] } template to be modified.

↓
denature

[]

forward primer attaches here

[]

backward primer attaches here.

Then, polymerase fills in the sequences.

DNA sequencing: given a molecule, we wish to find the nucleotides present in it.

DNA synthesis, making chains of nucleotides.

So, writing is synthesis.

reading is sequencing.

Challenges:

- consistency, durability (make it redundant)

- density - small area means how much info.

drawbacks?

- random access.

Background on DNA manipulation:

- selective DNA amplification with PCR.

- DNA synthesis

- DNA sequencing. (determining nucleotides in a DNA strand)

How? ✓

Polymerase.

each nucleotide emits a different colour. → see.



seq → template, primers, nucleotides, polymerase.

How will we create a DNA storage system?

reading writing.

• System overview

• Interface and addressing

• Access? System operations? See slides ^{oo}.

Synthesizer to do data to be stored.

synthesizes molecules & stored in a library

then while fetching

PCR thermocycler finds in a certain pool
(strand of interest is fetched).

given to sequencer, who decodes and gives it back.

when you're writing, you're writing objects.

How do you know where the data is present?

Basically, how do you address and retrieve?

(key, value) architecture (hash bucket jaisa again)

present in obj we're storing

since we're using primers for random access.

↑
primers can be mapped for hash func to retrieve ???

what to compute key on? → prime.

↓
put it in one of the pools based on key.

get $\Theta(\text{key}) \rightarrow$

put(key,value) → basically write.

so say foo.txt is the file.

see slides

for how read and write is done

becomes key

used for determining primer sequence encoded as high bits of addresses.

value

translate into nucleotides. → combine with key part (?) and store.

When you get key → if

↳ you're supposed to create a primer for that key.

now this primer needs to be embedded inside the object you need to store itself. that's why, we're combining it with the value part.

So primers are kinds

like addresses in the PCR

↓
it'll detect the string

having a particular sequence.

You can find only if while fetching primer is unique.

So, for each objects there should be a unique primer.

for mod 7
1 → } map to pool 1.
15 → }

now, if I do PCR with 1 primer, only 1 fetched.
so, random access.

Representing Data
↓

how to write in terms of nucleotides.

(0, 1, 2, 3) → (A, T, G, C).

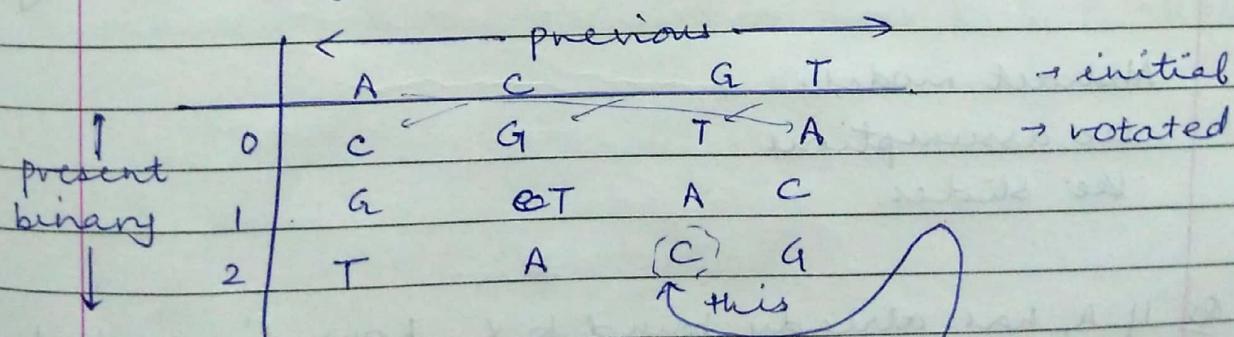
eg: 01110001 to base 4.
↓ ↓ ↓ ↓
1 3 0 1
C T A C

use base 3 to avoid errors. (sequencing errors).

↓

take binary, use Huffman coding and convert
into 3 kbp code.

a rotating encoding table



you've to create table kind from initial sequence
next is 2, previous G so see

Binary: 01 01 0000

Base 3 : G C

→ check.

what if 1?
what?

previous is G,

check.

Data Format.

higher end \rightarrow which post
address
lower end address \rightarrow "block in the post".

See slides.

Peptide Computing

(also under molecular computing)

seq. of amino acids

on peptides are specific recognition sites called epitope.

antibody

does every antibody have a unique epitope?

antibodies attach to these.

multiple antibodies can get attached to a single epitope (unlike DNA computing)

so if both present, who'll get attached?

based on binding power or affinity (higher).

Abstract model.

see assumptions

see slides.

Q: If A₁ has already bound to X, how do you get it apart?

use A₂, something with higher affinity to X.
A₁ nihal jaayega, A₂ lag jaayega.

- Given 2 multisets $G \& H$, can we find a subset quantity of X greater in G or H ?

Peptide creation for solving the problem.

4 antibodies $\rightarrow A_1, A_2, B, X$
 \downarrow
 not in $G \& H$.

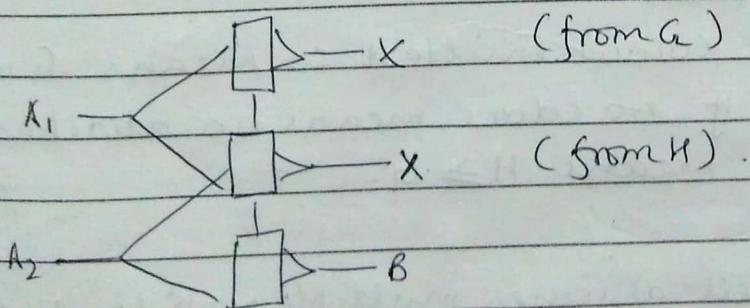
we have 3 recogn' sites.
 X has 2 disjoint sites.

A_1 attaches to peptide (not one).

see affinity $\rightarrow B > A_2 > X > A_1$.
 now, which antibodies will attach?

overlap for A_2, B ? check.
 A_1, X .

now, suppose X only on first site (upper epitope). Can A_1 join? No. Cuz A_1 ke do hain (and even A_2 ke do hain). So, X has to not be there for A_1 to attach.



labelled X means \rightarrow going to emit colour.

based on colour you can determine whether X from G or H more emitted or not.

consider

$$G = \{X\}$$

$$H = \{\dots\}.$$

X from

abhi will emit colour. Somehow
G has more Xs.

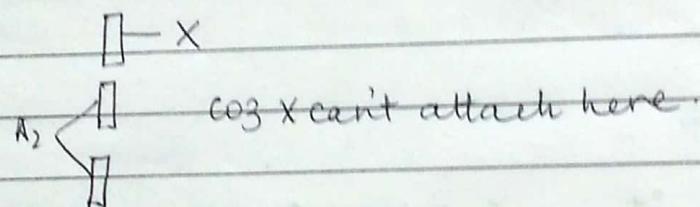
if there's no emission, H has atleast same Xs.
~~also tag along the~~.

Step 1. to prevent X from G getting attached to second epitope,
we put the antibody A₂ first.

Step 2. Then G is added.

$$\text{let } G = \{X\} \text{ and } H = \{\dots\}.$$

Now



Step 3. Now you've to remove A₂, so put B

Step 4. Add A₁, permanently attach to peptide.

* Lekin abhi A₁ can't.



A₁ used when H has more elements. We'll see.

5. Now, cross linker → covalent (permanent attachment).

6. Second set H added. ?

7. Labelled X. → colour emitted so means G more.

If no colour, means no attached,
means $H \geq G$.

for multiple Xs, will assume multiple peptides. That way
at least one peptide will emit or not emit a colour.

Also if $G > H$ colour

else ($H \geq G$) no colour.

Incremental Peptide computing

Hamiltonian & SAT

→ advantage over crude methods

coz you encode & process in

loops.

(build further
only if
contributing
to path)

traditional bio computing

has the disadvantage of synthesizing
all possible ~~paths~~ and then filtering
out the correct ones.

(amount \ggg ans)
of all generated.

Hamiltonian PP.

vertex i, epitope epi

$$v_1 \rightarrow e_{p_1}$$

$$v_2 \rightarrow e_{p_2}$$

so peptides are corresp. to paths

let source 1, end n

so, e_1 e_n } epitopes

$e_1 e_2 e_2 e_3 e_3 e_4 \dots e_n$
.
 e_1 .
 e_1 .
 e_n

we can have so many possibilities of peptides!

if e_1 is fixed, $\therefore (n-2)!$ permutation (doubly
duplicated permutation)
coz $e_2 e_3, e_3 e_2 \dots$

Now, we're to design antibodies for them.

Set A → all edges present in the graph
antibodies

Set B → ~~no~~ edge not in graph

Set C → entire set (for entire path peptide) (labelled).

If $e_2 - e_3$ exists,

will create peptide seq. $e_2 - e_3$ in set A

If $e_3 - e_4$ not exists

epitope, antibody $e_3 - e_4$ in set B.

also, $\text{aff}(B) > \text{aff}(C) > \text{aff}(A)$

see slides.

Now, In conventional,

add A first.

So wherever edges in graph, they'll attach there.
then B

(no edges there), there attached.

now C,

will remove A's,

emits colour \rightarrow that's the path you have!

(kyunki it'll replace the A's coz
of higher aff, so fluorescence!)

problem $\rightarrow (n-2)!$ too many.

So, in incremental,

encoding: pre_i x_i suf_i \rightarrow except start and end.

and every v.w. is unique.

if i-j edge, and epitope ij (suf_i, pre_j have
epitope mein),
it'll only link if edge is present.

Start & end to encoding

1st \rightarrow x₁ suf₁

nth \rightarrow pre_n x_n.

now, source : $\{S_1, S_2\}$

target : $\{S_2, S_3, \dots, S_n\}$.

now, if $S_1 - S_2$, antibody attaches.

$S_1 - S_3$,

;

;

now, for all antibodies attached, that becomes new source. and remaining in target.

So, we're growing only those which can become full path.

exception cond: $S_1 - S_2 - S_3 - S_2$

~~Reset~~ ↑

invalid HP.

so, how do we detect duplicate vertices?

take coloured epitopes \rightarrow if same colour, that means same vertex (there) more times.

See algo.

compare incremental and conventional

↓

↓

- n peptide
- (n-2)!
- less generations
- need to create all
- n-1 iterations
- one go.