(Q)

	1	2	3	4	5	6
R	10	4	8	12	6	10
B	4	7	10	3	15	7
G	9	2	8	1	7	5

Paint houses min cost such no 2 adj house of same

		Min(7,2)				
R	$10^{c=R}$	12^{G}				
B	$4^{c=B}$	6^{c} (Min(4,2))				
G	$9^{c=G}$	13^{c} (Min(4,2))				

dpR (i) → min cost to paint 1→i such that i^{th} house is Red.

dp B (i) →

dp G (i) →

$dpR[i] = dpR[i-1] + a[i][0]$

$dpR[i] = \min[dpB[i-1], dpG[i-1]) + R(i);$

$dpR[0] = R(0)$
$dpB[0] = B(0)$ } → Base
$dpG[0] = G(0)$

$dpB[i] = \min(dpR(i-1), dpG(i-1)) + B(i);$

$dpG(i) = \min(dpR(i-1), dpB(i-1)) + G(i);$

R	10	8
B	4	
G	9	

$ans = \min(dpR(n), dpB(n), dpG(n))$

Space → ~~6~~ maintain 6 variables.

(Q)

	1	2	3			
A:	10	7	8	12	18	5
B:	4	2	10	17	1	20

min time to produce all products.
if you switch from A-B or B-A there is an additional overhead of c. c = 2

	Min(7,4+c)				
10					
4	Min(2,10+c)				

$dp[i] = dp(i-1)$
dp(i) → ~~produced pro~~ min cost till production of (i)th machine

$dp(i) =$
$dpA(i) = \min(dpA[i-1], c + dpB[i-1]) + A[i];$

$dpB[i] = \min(dpB[i-1], c + dpA[i-1]) + B(i)$

$dpA[0] = A[0]$
$dpB[0] = B[0]$

$ans = \min(dpA(n-1), dpB(n-1))$

1 2 3 1 2

Kadane

Scanned with CamScanner

(Q) Find the max subarray sum.

```
int fs = INT_MIN, s = 0;
for(i=0; i<n; i++)
{   s += a(i)
    if (s > fs)
        fs = s
    if (s < 0)
        s = 0;
}
```

	1	2	3	4	5	6	7
$E_i \Rightarrow$	5	10	9	12	9	15	
Length →	1	2	3	4	5	6	7

find the upper bound
if found then replace
else insert

3 8 9

upr(4) = 8
replace 8

(Q) LIS

a_N: 5 10 8 9 3 12 . . . (Q) no. of increasing subsequences

```
dp[n];      dp(i) = LIS till i
dp[0] = 1
for( i=1; i<n; i++)
{   dp(i) = 1;
    for(int j=0; j<i; j++)
    {   if (A[i] >= A[j])
            dp[i] = max[dp(i), dp(j)+1)
    }
}
ans = max(dp)
```

```
dp(0)=1     sum=1
for(i=1; i<n; i++)
{   for(j=0; j<i; j++)
    {   if (A[i] >= A[j])
        {   sum += dp[i-1];
            dp[i] = max(dp(i), dp(j)+1)
            sum += dp[i-1]+1
        }
    }
    sum += dp[i];
}
```

(Q) Find the largest sub-Matrix sum.

5 10 8 9 3 12
1 2

```
if (A(i) >= A(j))
{   sum += dp(i-1)
    dp(i) = si
}
```

ds

like dsm

→ further optimization from n^2

Take an E array which contains smallest element which ends for LIS of length i

(Q) No. of increasing subsequences

$dp[i] \rightarrow$ # LIS till i^{th}.

$$dp[i] = \sum_{j=0}^{i-1} dp[j] + 1$$

$a[i] > a[j]$

else $dp[i] = 1$

2	3	
-1	-3	4
5	3	-2

$\rightarrow$

1	3	6
0	-1	6
5	7	12

$a[i][j] \rightarrow$ from $(0,0) \rightarrow (i,j)$

```
for (i=0; i<n; i++)
  {  for (j=0; j<n; j++)
     {
        s = p[i][j];  → sub-matrix sum from (0,0) to (i,j)
        a=b=c=0
        for (k=0; k<n; k++)
        {  for (l=0; l<n; l++)
           {  if ((k-1)>=0)
              {  a = p[i][(l-1)];
                 p[k-1][j];
              }
              if ((l-1)>=0;
              {  b = p[i][(l-1)];
              }
              if ((l-1)>=0 && (k-1)>=0)
                 c = p[(l-1)][(k-1)];
```

(i,j) (k,l)

	1	2	3
1			
2			
3			

$(1,2)$

(i,j)

(k,l)

```
for (y=i; y<=k; y++)
    for (z=j; z<=l; z++)
```

$val = S - a - b + c$

```
if (val > gS)
    gS = val;
```

2-D - Kadane

```
dp → colum prefix=1
for i<0 to N
{   for j<i to N
    {   for k<- 0 to M
        {   ans += dp[j][k] - dp[i-1][k]
            gs = max(ans, gs)
            ans = max(0, ans)
        }
    }
}
```

$O(n^3)$

(Q) arr: 10, 3, 15, 8, 5, 17

pick either end points

Two player game.

Find the winner (max sum)

$l = n$, $r = m$

$n-1$

~~dpA[i] = max~~

~~dp(i)(j) = max score of player~~

1000

1800

10000

l, r

$dp(i)(j)$ → max score in (i, j)

$$\text{Max}\begin{cases} a(i) + \text{Min}(f(i+2, j), f(i+1, j-1)) \\ a(j) + \text{Min}(f(i, j-2), f(i+1, j-1)) \end{cases}$$

$i == j$ → $V(i)$

~~abs(i-j) == 1~~ → $\max(V_i, V_j)$

$(j == i+1)$

We are choosing min so that the other player gets min

dice,

Domino,

Submatrix,

Painting House

2-ST

2→1B

1000 0000000000000

1 1 256 64 16 4 1

1, 2, 4

1111

1000

101001

101

110

dp[0][2] = s = s+1

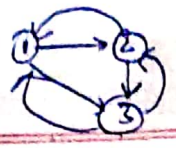
1 1 1 1 1

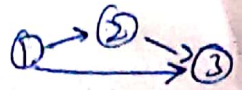
1 2 4 8 16

$n \geq n$

S 1 S

Scanned with CamScanner





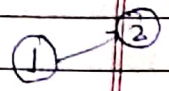
1 2 3 1 3 2 1

1 1

① → FC → same

②

④



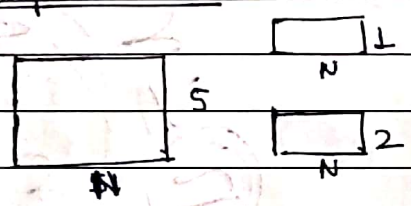
1 0 0 0

0 0 0 1

dp(0)=0
dp(1)=0
dp(2)=0
dp(3)=1
dp(4)=1

0 00 111
0 000 111

• Arranging Dominoes



dp[i] → no. of ways to fill Binary strings with no adjacent 1.

n=1 00 1

n=2 00 01 10 10 01

00 01 10

n=3 001 100 101 101

010

a → ends with 0 b → ends with 1

a[0]=1 b[0]=1

0 01 10

00

n=1 0 1

n=2 00 10

01

n=3 000 ~~010~~ 101 010

001

10 22 9 33 49 50 31 60

1 1 1 1 1 1 1 1

dp Questions cfg

① Paths to reach (0,0)

② Max Sum by breaking into n/2, n/3, n/4

③ Tiling Problem.

④ Ugly number → Find nth num which 2, 3, or 5 as prime factor.

⑤ Reach a given score → 3 or 5 or 10

⑥ Optimal strategy game.

④ ⑦ Skip Game (skip tasks but can't skip 2 consecutive)

⑦ Longest alternating subsequence → take two dp's (greater and lower)

⑧ Coin change Problem → dp → DFS

⑨ Friends pairing problem

↳ 1) doesnt pair → f(n-1)

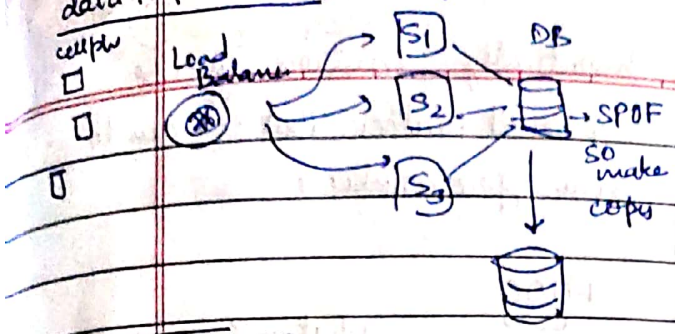
2) pair with (n-1) → (n-1) * (f(n-2))

⑩

Date
Page

Scanned with CamScanner

Distributed consensus and data Replication strategies

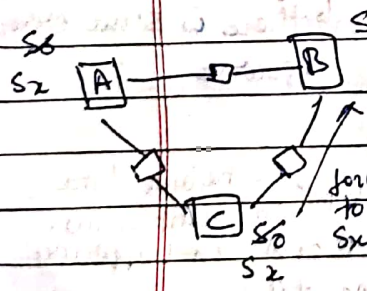


Raid Tech ??

Replica → Syn, Async } depends on the application we are handling.

- Only Master allows read op and slaves copies the master. So, write can only be done in master → (limitation).
- ∴ We can have, Master-Master architecture, where 'write op^n' can happen at slave also.

A problem lies b/w communication b/w the Masters. (if fails) it can lead to Split Brain Problem.

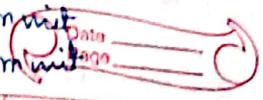


assumption
- if C fails A-B remain intact
- if A-B fails C remains intact.

so, a validator always exists

This is a distributed Consensus, multiple nodes are agreeing to a common value.

- 2PC → 3PC
- 2Phase commit
- 3 phase commit

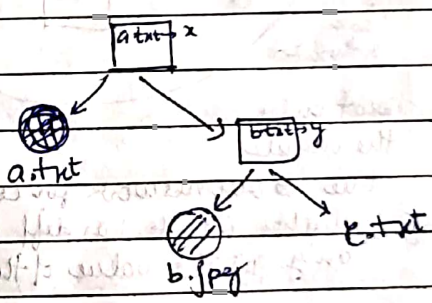


Some times we need to have long Transactions like Food ordering. You ordered food and you it doesn't actually withdraw the money, rather it locks it. If the restaurant confirms it, then OK else rollback. This saves the problem of settlement which take 2/3 days.

SAGA → handles this long transaction

Git's internal DS : Merkle Trees

① Git ② DynamoDB ③ BlockChain

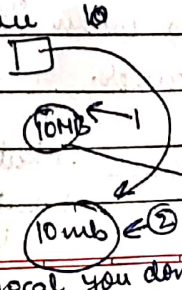


So, what it does, store the pointer of the file through a hash fn. h(content) = a.txt.

So, now if the content is changed these changes will give new hash. So, these must be accounted for.

- changes are propagated up till the root.

you have



you change 1 line → garbage.

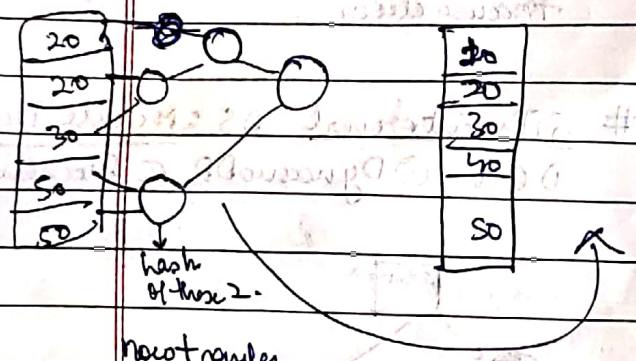
now at local you don't care. but while pushing to remote, git compresses those

(8) We want to have an exact copy of all values in the range 0–9 from the old node into the newly added node. But the data is constantly updated in the old node.

use Merkle Trees.

→ Keep hashes of all ranges in a tree format and ~~com~~ propagate values only if hashes are different.

→ no need to copy the unchanged data again.



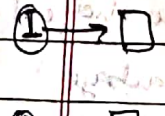
now transfer the merkle tree to our network for comparison (whichever node has diff hash) only get the value of the node)

Hyperloglog Algorithm

Problem → count the unique users connected to your web site at any given point of time?

Basic Algo Time O(1)
Space O(n)
Using Hash Map.

but challenge comes when it's in a distributed manner.



count=2 → but only 1 unique user.

Date
Page

theoretically, solution can't be improved further. But we can do some approximation.

1000
x zero → what's the probability of getting this no.

$\frac{1}{2^x}$.

So, we can say, we have hit 2^x requests →

but it's not always the case.

11000
00000
10011
10100

5 zeros $= 2^5$
$= 32$ req
but we have only 4.

So, now use hashing.
now take the average bucket size. say (y).
2^y → is the total no. of requests (nearly 2^y)
↳ there is some error of course.

Raid Technology

- Raid 0 → striping → distribute data on multiple HD. and R or W can happen on any one of them.
  ↳ data is highly insecure - (can be lost) → but double speed
- Raid -1 → Mirroring → copy HD1 → HD2. Same speed but data lost is compensated.
- Raid 10 → Hybrid of above two

Scanned with CamScanner

(8) subset sum → decision problem using dp (variation of knapsack)

dp[n+1][sum+1].

for (i ← 0 to n)
    dp[i][0] = true → 0 sum always true.

for (i ← 0 to sum)
    dp[i][0] = false → empty set atum false.

for (i ← 1 to n)
    for (j ← 0 to sum) → if guess num is >
        if (j < a[i-1])
            dp[i][j] = dp[i-1][j]
        else
            dp(i,j) = dp(i-1, j) ||
                dp(i-1, j - s(i-1));

return (dp(n)(sum));

we can optimize further (space) to 2×sum because we only need just the previous row inf.

dp[2][sum+1];
    if (i==0)
        dp[i%2][j] = true
    else if (i==0)
        dp[i%2][j] = false
    else if (a[i-1] <= j)
        dp[i%2][j] = dp(i-1%2, j) ||
            dp(i-1%2, j - a(i-1))
    else
        dp(i%2, j) = dp(i-1%2, j);

return dp(n%2)(sum);

• longest alternating subsequence

dp[i][0] → LAS till i where i is >
dp[i][1] → LAS till i where i is <

INPUT	10	22	9	33	49	50	31	60
0	1	1	1	4				
1	1	1	2	1				

INTMD	10	22	9	33	49	50	31	60
	1	2	2	3	3	1	1	1
	1	1	2	1	1	1	1	1

if (a(i) > a(j)) → ith element is greater so length would come from small (i.e las(1))

if (a(i) < a(j)) → ith elem is smaller so length would come from large (i.e las(0))

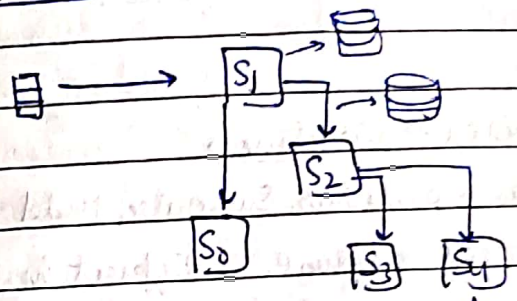
• Gold Miner

1	3	3
2	1	4
0	6	4

→

1	4	7
3		
0		

• Publisher Subscriber Mode



• S_1 sends two messages after process
• then S_2 sends two mess to S_3 S_4 after processing.

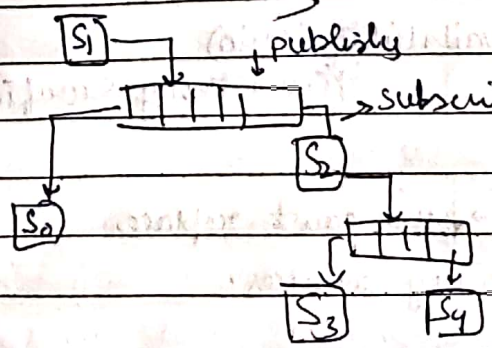
But what if S_4 fails, then client would get Time Out.

Again process starts, so again processing is done by S_1 & S_2 which may lead to data inconsistency and also Time Overhead.

→ So a better way is to have a message queue (kafka) which will ensure that the message will reach eventually and client gets success.

• S_1 & S_2 will return success after their message passed to queue.

• Loose transaction



× data consistency (on funding transaction) not on finance → no way

- data inconsistency because this model is having transaction across services.

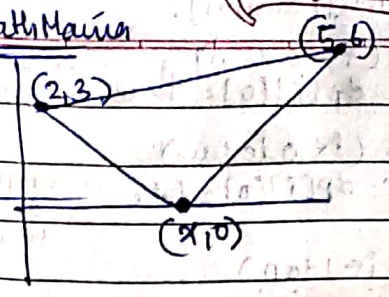
say → S_1 → does bank charge
S_2 → fund transfer

↓ S_1 & S_2 must have to be atomic

else too much confusion.

Date
Page

MathMania



$(2-x)^2+3^2+(5-x)^2+6^2$

$-2(2-x)-2(5-x)=0$

$-4+2x-10+2x=0$

$4x=14$

$x=\frac{14}{4}$

$x^{1/2}$

$\sqrt{(2-x)^2+(5-x)^2}+a+b$

$\frac{1}{2}x^{-1/2}$

$\Rightarrow \frac{1}{2\sqrt{x}}$.

$2+36$

$\frac{1}{2\sqrt{(2-x)^2+(5-x)^2+a+b}}(-2(2-x)-2(5-x)+45)$

$45-14$

31

$-4+2x-10+2x+45$

$4x=31$

$x=31/4$

Event driven Architecture →

basis → publisher Subscriber Model

- Request response is different in a way because in this architecture services are concerned about events. Any service can consume that event and then publish in the Event Bus.
- easy Rollback
  - ↳ Service takes the event and store in local dB with a timestamp then whenever a buy comes, easily identifiable.
- Easy Replacement of services
  - start the new service and play in all the events from local dB and the event bus then it will become consistent with others.
- Availability (OVIO)
  - ↳ (Consistency sacrificed)

- dis → you can't replace gateway services.

(2)

$-2(x_a-x)-2(x_f-x)+y_a^2+y_f^2=0$

$-2x_a+2x-2x_f+2x+y_a^2+y_f^2=0$

$x=\frac{(y_a^2+y_f^2)-(2x_a+2f_x)}{4}$

$\frac{1}{2\sqrt{y}}(2(x-2)+2(x-5)+36)=0$

$2x-4+2x-10+45$

$4x=-31$

$x=\frac{-31}{4}$

$2x-4+2x-10$

$4x=14$

$x=14/4$

45

$x^2=2|x|$

$(2-3.5)^2+(5-3.5)^2+9+36$

$2|x-2|+2|x-5|=0$

$\Rightarrow |x-2|+|x-5|=0$

$1.5^2+1.5^2+45$

① $2-x+5-x=0$ $x=3.5$

② $x-2+5-x\neq 0$

③ $x-2+x-5=0$
$x=3.5$

Scanned with CamScanner

(Q) Count subsequences of the form $a^i b^j c^k$

Date
Page

a b c a b c

1	1	1	1	2	2

a b c a b c

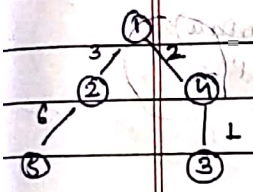
1	1	1	3		

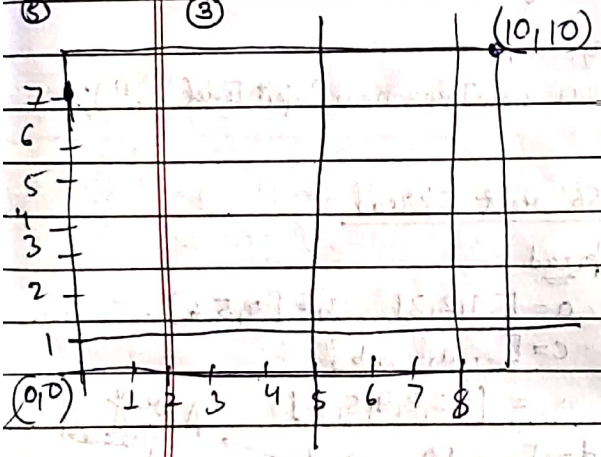
$S = \frac{48}{7}$

1→1
2→2
3→1
1,2 → 4
2,1
3

1 3	1
2 4	1
3 5	2
4 6	3
5 7	4

• Super Maximum Cost Queries





xc = 0, 10	xd = 10
yc = 0, 10	yd = 10

is Pal → 2D dp

l=1 → dp(i,i) = true

l=2, if s(i) == s(i+1) → dp(i,i+1) = true

for l=3 to n.

i = 0; i<n-k+1.

end = j = i+k-1;

start = i

if s(i) == s(j)

↳ means if (i+1, j-1) is a pal. then this is also

if (s(i) == s(j) && dp(i+1, j-1)

dp(i,j) = true.

(31)

Flip array (30).

(a, i+1, s), (a, i+1, s-a(i))

if (s<=0) (s>=0)

return min(m, s);

#Load Balancer

• Hardware LB

• Software LB → HA Proxy.

Algo → • Round Robin
• Round Robin with weighted server.

• Least connection

• Least Response Time

• Source IP Hash / URL Hash.

#Cache

• Time Locality
• Space

Types

① Application level cache

↳ each server will have it's own cache but say value of a is cached but the

→ Request goes to another server → No use of cache.

② Distributed Cache.

Request will hashed to their particular server so that the previous issue is addressed.

③ Global cache → First check in global cache then process the request.

Scanned with CamScanner

Cache Invalidation

① Write through
- update cache and DB at the same time.

② Write back
- update cache (giving customers greater imp) then update db in bunch at regular interval.

③ Write around
- update db first then cache.
if DB is updated then there would be cache miss.

Cache Eviction
- → FIFO
- → LIFO
- → LRU
- → MRU

JS

var, let

var a=5
var a=6 ✓ ⇒ let a=5
let a=6 ✗

var and let follow variable Hoisting → declaration move up.
but definition stays there.

ES6 → arrow functions ↓ u2

function sum(a,b){	let sum=(a,b) ⇒ {
return a+b;	return a+b;
}	}

Classes in JS

```
function Todo (topic, due){
    this. topic = topic
    this. dueDate = dueDate
    this. Completed = false
}.

Todo.prototype. Completed = function (
        this. completed = true).
```

```
var todo = new Todo (' ', ' ')
    todo. completeTodo();
    console. log (todo).
```

{ES6 class}

```
class Todo (){
    constructor (topic name, date) {
        this. name = name;
        this. Date = date;
        this. completed = false;
    }
    markCompleted () ⇒ this. completed = true;
}
var todo = new todo(' ', ' ');
```

• inheritance in ES6 classes

```
class projectTodo extends Todo {
    constructor (name, dueDate)
    {   this. type = 'project';
        super(name, due);
    }
}.
var newTodo = new ProjectTodo('', '');
```

• ES6 rest spread

spread

```
a = [1,2,3]    b = [4,5,6].
c = [...a, ...b]
  = [1,2,3,4,5,6]
d = [...b, 10, ...a, ].
```
→ Array spread

```
var a = { name: 'abc', date: '10 Aug' }.
var b = { name: 'xyz', type: 'proj' };
var c = [...a, ...b] .
```
→ object spread

Here b will override.

```
c = { name: 'xyz'
      date: '10 Aug'
      type: 'proj' }.
```

Scanned with CamScanner

• difference b/w rest op and spread op.

when ... is used during defining of a function → rest

when ... is used outside or assignment then it's spread.

Spoj – Seinfeld

} → and s is empty → +1

① { → push to stack

②

③ {{{}}

{{{{{}}{}} →

• ES6 Destructuring Assignment.

var todo = { name, due, type, complete }.

var [name, due] = todo;
↳ destructuring.

var a = [a, b, c, d, e].

• get e.

var [x, y, , z] = a.

z = a[4] = e.

var [x, ...z] = a

x = a(0)

z = [b, c, d, e].

• odd palindrom

xabcx → 12221

a b a c x y x

| 1 | 2 | 1 | 1 | 1 | 2 | 1 |

a b a a x y x a

| 1 | 3 | 2 | 1 | 1 | 1 | 1 | 1 |

4 3 2 2 2 2

a b a a x y x a

| 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |

1 1 1 1 1 1

1 1 1 1 1

1 1 1

1 2 2 2 3 4 3 1

• Map Filter Reduce :-

0 1 2 3 4 5 6 7 8 9 10 11 12

3, 5, 7, 9, 20, 25, 30, 40, 55, 56, 57, 60, 62

1, 4, 7, 11, 14, 25, 44, 47, 55, 57, 100.

405

429, 426, 421, 414, 385, 360, 330, 290, 235, 179, 122, 62

365, 364, 360, 353, 328, 303, 259, 212, 157, 100

342

3, 5, 7, 9, 20, 25, 30, 40 55

0 1 2 3 4 5 6 7 8 9 10 11 12

429, 426, 421, 414, 405, 385, 360, 330, 290, 235, 179, 122, 62

0 0 365, 364, 360, 353, 342, 328, 303, 259, 212, 157, 100.

57

3

1	2	3
4	5	6
7	8	9

1	4	7
2	5	8
3	6	9

Scanned with CamScanner

3x3

9	5	5
3	3	3
3	2	1

4x3

12	8	6
5	5	3
3	3	3
3	2	1

4x4

16	12	6	7
4	5	5	3
3	2	3	3
3	3	2	1

5x4

20	14	10	8
7	7	5	5
5	3	3	3
3	3	3	3
3	3	2	1

5x5

25	13	9	7	9
5	7	4	4	5
5	3	3	3	3
3	3	2	3	3
3	3	3	2	1

m (m x k)

Date
Page

k

7 4

k=2 1 2 3 1 2 3 4

1 2 3 4 3 2 1

1 2 3 4 1 2 3

n x m

n x m m x n

k k

k = 3 4

1 2 1 3

1 2 3 1

3

your P

C_i P_i H Horeach.

10 k outside temp

16

10^5 1 2 3 4 5

k=3

k=3 1 2 1 3 3

1 2 3 3

1 3 2 1 3

4/3 1 7/3

m = 1 ✗

m = 2 ✓

2 3 2 1 3

m

m = 1 ← q

m = i

i x k

1 2 3 1 2

1 1 1 2 2 2 3

1 2 1 1 2 2

1 2 3 1 2

19 20

20 20 20 20 10

21 19 16

22 18 17

19 18

1 2 2

5 2

3 4 6 3 4

3 3

3 4 3 4

k = unique

$^nC_0 + {}^nC_1 + {}^nC_2 + {}^nC_3 + {}^nC_4$

$^{n+1}C_1 + {}^{n+1}C_3 + {}^nC_4$

Scanned with CamScanner

21, 22, 23

20, 20

Date
Page

x x x+7 y 3

• get hash f^n

• Building Bridges Hard

(1,3) (2,4) (3,5), (4,6) (5,1), (6,2)

22, 22 23, 23

20 22

(1,3)

(2,4)

(3,5)

(4,6)

1 2 3	2	1
1 3 6	8	9
9 8 6	3	1

R = 7 | R = 6

R = 6 | R = 7

R = 8

1011
0111
1100

1001 → 9
10110
11111

1011
1011
0

• KAOS

avon — avol, 2

lova — caion, 3

novae — ecnus, 4

sunc — nora, 1

1001
00111001

00000100
11111100
11111008
7654321

128
64
32
16
8
268

Surface area of cone

Lateral → $\pi r l$

Base = πr^2

$\pi r(r+l)$

$\pi r(r+\sqrt{h^2+r^2})$

$\pi(\sqrt{6^2+4^2})$

$\pi(\sqrt{36+16}) = 22.6543$

101
111

000100
111111
111011
3148210

32
16
8
3
79

$\frac{1}{3}\pi r^2 h$ →

$\frac{6\times5\times4}{2\times4}$

$\frac{6\times5}{2\times4}$

$\frac{6\times4\times5}{15}$

1001
0111
1110
8424
814
1011

1001
1011
0010

$^6C_1 + {}^6C_2 + {}^6C_3 + {}^6C_4$

$\frac{R}{r} = \frac{H}{h}$

6 + 15 + 20 + 15

$r = R \times \frac{h}{H}$

$^2C_1 \times {}^2C_1$

2×1

Scanned with CamScanner

$Z_n + Z_{n-1} - 2Z_{n-2}$

$(1^1+2^1+\ldots n^n) + (1^k+2^k \ldots n^k)$

$+ (1^1+2^1 \ldots (n-1)^{n-1}) + (1^k+2^k \ldots (n-1)^k)$

$-$

$2[1^1+2^1+\ldots(n-2)^{n-2} + 1^k+2^k+\ldots(n-2)^k)]$

$\Rightarrow 2[1^1+2^1+\ldots(n-1)^{n-1} + 1^k+2^k+\ldots(n-1)^k] + n^n+n^k$

$-2[1^1+2^1+\ldots(n-2)^{n-2} + 1^k+2^k\ldots(n-2)^k]$

$\boxed{(n-1)^{n-1} + (n-1)^k + n^n + n^k}$

(Q)

$(4)^3 + 4^1 + 5^5 + 5$

Binary Exponentiation →

① Computation of large exponents modulo m.

② Effective computation of Fibonacci numbers. → Imp

↳ Matrix Exponentiation

③ Find no. of Path-lengths of k in a graph.

↳ Raise the adjacency Matrix to power k.

Fibonacci using Matrix Expo →

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$$

0 1 1 2

calculate this using Binary Expo

• change values using initial terms of given Fibo

$$\begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix} \begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix}$$

$$= \begin{bmatrix} f_{00}m_{00} + f_{01}m_{10} & f_{00}m_{01} + f_{01}m_{11} \\ f_{10}m_{00} + f_{11}m_{10} & f_{10}m_{01} + f_{11}m_{11} \end{bmatrix}$$

Suffix Arrays

Suffixes → from the end of a string.

• Suffix Array → contains all the suffixes in sorted manner

Camel

0 → camel
1 → amel
2 → mel
3 → el
4 → l

1	amel
0	camel
3	el
4	l
5	mel

↳ this is the actual suffix array (sorted indices)

• LCP array

as it stores how many characters (from start) are common in 2 adjacent sorted suffixes.

(Q) Find unique Substrings.

Text → AZAZA

→ Suffixes AZAZA, ZAZA, AZA, ZA, A (0, 1, 2, 3, 4)

SA	LCP
4 → A	0
2 → AZA	1 → in AZA and A there is 1 repeat
0 → AZAZA	3 → in AZAZA and AZA there are 3 rep subst
3 → ZA	0
1 → ZAZA	2

So total unique substrings

$$\frac{n(n+1)}{2} - \Sigma(LCP)$$

(Q) Find Longest common substring given n strings → find LCS for exactly k strings.

Imp (use many DS)

Scanned with CamScanner

① Build Suffix array ↦ concatenate all strings using a delimiter (different for all strings)
(creates suffixes then sort them).

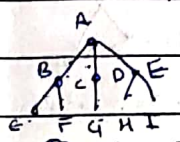
② . create LCP array.

③ use sliding window of length k. (if that window contains suffixes of k different strings then query in LCP array (using RMQ (Seg Tree)). that will be the answer. (after omitting the 1st element) (because LCP[0] is undefined)

0 → ababa	
1 → baba	
2 → aba	
3 → ba	
4 → a	

→

4	a
1	aba
3	ababa
0	ba
2	baba



3 → 3
2 → 1
1 → 5

$v_1+v_2+v_3=n$

⑤. $3v_1+2v_2+v_3 = 2E$

$\Rightarrow 3v_1 + 2(n-v_1-v_3)+v_3$

$\Rightarrow 3v_1 + 2n - 2v_1 - 2v_3 + v_3 = 2E$

$\Rightarrow v_1 - v_3 = 2(E-n)$; $v_1-v_3 = 2(n-E)$

$v_1-v_3 = 2(n-n+1)$
$v_1-v_3 = 2$

$3-5 = 2(8-9) = 2$

3 → 1
2 → 3
1 → 3

$3-1 = 2(6-7)$
$2 = 2$ ✓

(a) Longest Repeated Substring.

① Create Suffix array.

② Create LCP array.
get the max in LCP array. (that would be length) and the common chars starting from 0 till max would be that string.

open. Kattis

Questions → ① dvapat
② Substrings
③ lifeforces.

(c) general d-degree graph.

$v_1 = 1$ deg, $v_2 = 2$ deg, $v_{d-1} = d-1$, $v_d = d$ deg

$v_1 + 2v_2 + \dots (d-1)v_{d-1} + d(v_d) = 2E$

$v_1+v_2+v_3 \dots v_{d-1}+v_d = n$

$v_1, v_2, \dots v_n$.

⑤. n

deg of each $\geq \frac{n-1}{2}$.

$\Sigma \deg(v_i) \geq \left(\frac{n-1}{2}\right) n$.

let G be disconnected.

S be one of the component

$x \in S$ → neighbour of x is atleast $= \left(\frac{n-1}{2}\right)$

$|S| \geq \left(\frac{n-1}{2}\right) + 1 \Rightarrow |S| \geq \frac{n+1}{2}$.

now examine V-S.

$|S| \leq n - \left(\frac{n+1}{2}\right) = \frac{n-1}{2}$.

$|S| \leq \frac{n-1}{2}$

$n = 2$

$\frac{n(n-2)}{2} = 2(n-1)$

$2(v_1+v_2 \dots v_d - 1) = v_1 + 2v_2 + 3v_3 + \dots dv_d$

$2v_1 + 2v_2 + 2v_3 \dots 2v_d - 2 = v_1 + 2v_2 + \dots dv_d$

$12 = 2x$

$v_1 - v_3 - 2v_4 - 3v_5 \dots - (d-2)v_d - 2 = 0$

$v_1 - (d-2)v_d = 2 + \sum_{i=3}^{d-1} (i-2)v_i$

Scanned with CamScanner

n is even

$\frac{n-2}{2}$

$\delta = \left(\frac{n-2}{2}\right)$

deg = 0

atleast deg $\geq \frac{n-2}{2}$

$\frac{4-2}{2}$

sum of deg $\geq n \cdot \frac{(n-2)}{2}$

$2(n-1)$

$n\left(\frac{n-2}{2}\right)$

$\frac{10 \times 8}{2} = 40$

$2(9) = 18$

Two comp. H, K.

$x \in H$.

$|H| \geq \frac{n-2}{2} + 1$

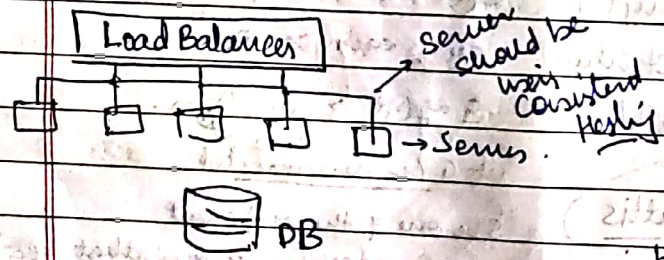
$|H| \geq n/2$.

$|K| \leq n - n/2$

$|K| \leq n/2$

disconnected

System Design I

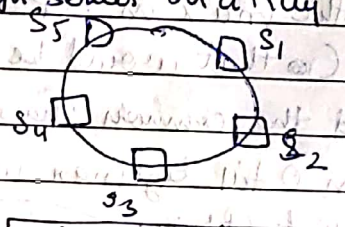


- You want to scale up.
  You have got multiple machines
  But DNS returns 1 ip only.
  So, you need Load Balancers to send the client to the correct server.

- How to decide which Server to choose
  → Load Balancing Techs
  - Round Robin
    (what would happen if some k^{th} server become slow?)
  - Least Response Time
    (send the req. to those servers with LRT)
  - Heartbeat
    (LB maintains a list of healthy servers)
    (This is a common practise of maintaing healthy hosts with them)
  - Weighted Round Robin (depending upon the power of servers).

(Q) Can you share the load b/w varying servers → Yes, but it's a better approach not to because Network hops must be minimized. RAM → HD → Same Cluster → Diff Data Center

- Consistent Hashing
  Assign Servers on a Ring

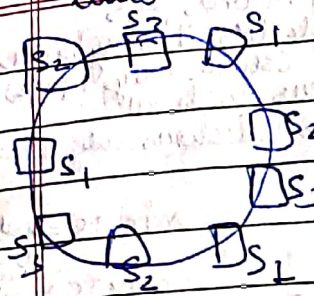


→ sorted array

Hash the user and find the just bigger hashed server (BS).

- Cascading Failure
  Say S_2 → dies → then S_3's load will be doubled
  Say S_3 dies → then S_4's tripled
  ↳ So each server will start dying one by one.
- Removing a server → Cascading Failure
- Adding a server → uneven load.

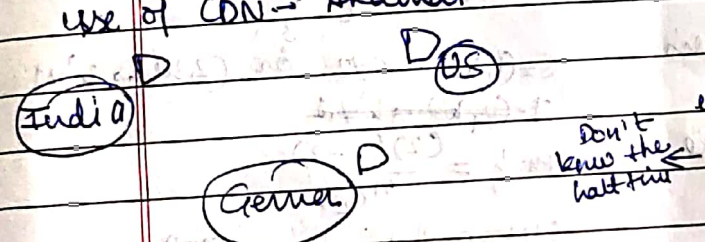
• Variant of consistent Hashing
create virtual replicas of the servers
i.e hash the servers multiple times



Physically they are same but the assignment can happen at multiple times.
(More Hashes → More distributed)
↳ (Multiple Token Servers)

System Design - II

use of CDN → Akamai



These CDN will keep the static content (images, videos) near to the user. Only the dynamic parts will need to be fetched from the server.

matches the regex and if matches then serve through the CDN

• cloudfront and Akamai has all the resources public. Meaning → anyone can view the content but guessing the URL is very difficult so until and unless someone gives you the URL, you can't access.

• CDN is given a TTL after which the resource is destroyed.

• client caching → the browsers are intelligent and cache the static data. whenever, you refresh the page, it will show you cached data, that's why sometimes you need to hard refresh the browser.

• Use case → Test data is changed for a particular problem → How would you make the system go into a consistent state.

(i) Halt the queries and update DB. ← Don't know the halt time
(ii) Cache - auto refresh
↳ use TTL for like 5 min
(then delete the cache data and fetch updated data from DB.)
But for that 5min window system will be in inconsistent state.
(iii) update all cache servers then update dB
(But what would happen if some server didn't respond (will you keep trying))

Caching Mechanism
↳ these Mech don't work on server cache.
It works on Global Cache → 1C, 1DB

① Write through
→ C
→ DB
2 writes
Always consistent
Success → only if both are completed
failure → if any one of them fails.

Scanned with CamScanner

• Write back → customer is more up → then update DB.

O → C → client

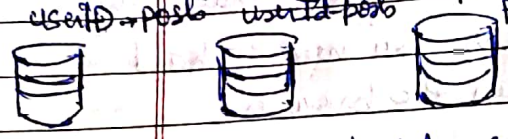
• Write Around

O → DB → update cache

Cache is inconsistent for some time.

FB newsfeed

~~usrID → post~~ ~~usrId post~~ usrID post



Find the newsfeed for a particular guy.

→ gets the user Id and it's post

→ get all the friends and their posts (say 2000 friends, 2000 DB queries) because the friends may not lie on same machine

{ab}
{ac}
1

• We can think like we don't need Immediate Consistency but Eventual Consistency would be fine.

Keep a cache (Big) which stores newsfeed of users Globally.

So, now you can query that cache to see if any of your friends are there or not.

Intermediary Global Post (2-3 hours of post arou to GiB).

DB

Post content

DA

CH-3 → Similar Items

• detecting Mirror pages. (Google does it very better (shows only 1 page))

• Google News finds all such articles but shows only 1 instance

$3/4 - 1/3$

(Q) → $U = n$

$S = m$, $T = n$.

→ $\frac{S \cap T}{n+m} = \frac{m}{n}$

min JS ⇒ when $U = sum$

max JS ⇒ when $U = (sum - int)$

S and T each with size m.

① size = 1

$SIM(S,T) = \begin{cases} 1 & P = 1/n \\ 0 & \text{others} \end{cases}$

② size 2.

$SIM(S,T) = \begin{cases} 1 & P = 1/{}^nC_2 \\ 1/3 \end{cases}$ (out of ${}^nC_2$ pairs)

$S = (1,k) \to k \neq 2$ or $(2,k) \to k' \neq 1$

~~$T = (2,k') \to k' \neq 1$~~

$SIM = 1/3 = \frac{(2)(n-2)}{{}^nC_2}$.

~~$SIM = 0 \to$~~ $1 = \frac{1}{{}^nC_2} - \frac{2(n-2)}{{}^nC_2}$.

③ size = 3

$SIM = \begin{cases} 1 \to P = \frac{1}{{}^nC_3} \\ \frac{1}{2} \to \frac{3(n-2)}{{}^nC_3} \\ \frac{1}{5} \to \frac{3}{{}^nC_3} \cdot 3\binom{n-3}{2} \end{cases}$

$E = \frac{1}{{}^nC_m} \sum_{j=0}^{m-1} \frac{m-j}{m+j} \binom{m}{m-j} \binom{n-m}{j}$

m size

$SIM = \begin{cases} 1 = \frac{1}{{}^nC_m} & P = \frac{1}{{}^nC_m} \\ \frac{m-1}{m+1} & P = \frac{1}{{}^nC_m} \binom{m}{m-1} \binom{n-m}{1} \\ \frac{m-2}{m+2} & P = \frac{1}{{}^nC_m} \binom{m}{m-2} \binom{n-m}{2} \\ \vdots \\ \frac{m-(m-1)}{m+(m-1)} & P = \frac{1}{{}^nC_m} \binom{m}{m-(m-1)} \binom{n-m}{m-1} \end{cases}$

abc abc
ab bc

$n-2+1$

3/

• similarity b/w minhash and Jacc.

$x \to$ both 1
$y \to$ 1 and 0
$z \to$ both 0.

$\Rightarrow SIM = \frac{x}{x+y}$

• Get the signature Matrix.

	S_1	S_2	S_3	S_4	$x+1 \% 5$	$3x+1 \% 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

	S_1	S_2	S_3	S_4
h_1	∞	∞	∞	∞
h_2	∞	∞	∞	∞

row 0 $h_1(0)$ and $h_2(0) = 1$ and S_1 & S_4 are 1

1 0 0 1
1 0 0 1

row 1 (2,4)
S_3 is 1

1 0 2 1
1 0 4 1

row 2 → {3,2}
S_2 and S_4.

1 3 2 1
1 2 4 1

{1<3}
{1<2}
↓
min values.

row 3 → {4,0}
S_1, S_3, S_4

1	3	2	1
0	2	0	0

row 4 → {0,3}
S_2

1	3	0	1
0	2	0	0

final signature Matrix.

	S_1	S_2	S_3	S_4	$2x+1 \% 6$	$3x+2 \% 6$
0	0	1	0	1	1	2
1	0	1	0	0	3	5
2	1	0	0	1	5	2
3	0	0	1	0	1	5
4	0	0	1	1	3	2
5	1	0	0	0	5	5

	S_1	S_2	S_3	S_4
h_1		1		1
h_2		2		2
h_3		2		2

h_3
2
1
0
5
4
3

row(0) → {1, 2, 2}
S_2, S_4.

row(1) → {3, 5, 2}
S_2

∞	1	∞	1
∞	2	∞	2
∞	2	∞	2

row 2 → {5, 2, 0}
S_1 S_4

5	1	∞	1
2	2	∞	2
0	1	∞	0

row 3 → {1, 5, 5}
S_3

5	1	1	1
2	2	5	2
0	1	5	0

row 4 → {3, 2, 4}
S_3, S_4

5	1	1	1
2	2	2	2
0	1	4	0

$\frac{2}{6} = \frac{1}{3}$

$\frac{1}{6}$

row 5 → (5, 5, 5)

S_1 → | 5 1 1 1 | / | 2 2 2 2 | / | 0 1 4 0 |

check the values of

	1,2	1,3	1,4	2,3	2,4	3,4
col/col	0	0	1/4	0	1/4	1/4
sig/sig	1/3	1/3	2/3	2/3	2/3	2/3

check the sig matrix

sig/sig → do parallel matching
$\frac{\text{total match}}{\text{Total hashes}}$

col/col → $\frac{\text{count matches}}{\text{Total docs}}$

Scanned with CamScanner

LSH

- Prob. that the signatures agree in all rows of 1 band $\rightarrow s^r$
- Prob. that sig disagrees in atleast 1 row of a particular band $\rightarrow 1 - s^r$
- disagrees in each band $\rightarrow (1-s^r)^b$
- Prob. that agrees in all rows in atleast 1 band $\rightarrow 1-(1-s^r)^b$

threshold $= \left(\frac{1}{b}\right)^{1/r}$

r integer to P. to from bottom, Three r

1	2	1
2	3	2
9	3	0

b_1 r row
b_2
b_3

for each band we have have a hash f^n that takes r integers and hashes them

- get the sig matrix
- divide into band r.
- apply LSH Technique

- Analysis of Bloom Filter

Darts $\rightarrow$ x targets and y darts

Prob of false positive

$$\left(1-e^{-km/n}\right)^k$$

$k \rightarrow$ hash f^n

$m \rightarrow$ set of key values

$n \rightarrow$ bit array length

$n = 8$ bill

$m = 1$ bill

$k = 3$

$$\text{fal } P = \left[1-e^{-\frac{km}{n}}\right]^k$$

$$= \left[1-e^{-\frac{3}{8}}\right]^3$$

$$\text{fal } P \text{ with } k=4 = \left[1-e^{-\frac{1}{2}}\right]^4$$

(Q) n, m, and k.

$$f = \left(1-e^{-\frac{km}{n}}\right)^k$$

$$\frac{df}{dk} = k\left[1-e^{-km/n}\right]^{k-1} \cdot \frac{km}{n} e^{-\frac{km}{n}}$$

$$\frac{df}{dk} = 0$$

$$k\left[1-e^{-km/n}\right]^{k-1} = 0.$$

$$e^{-km/n} = 0$$

$$-\frac{km}{n} = 0.$$

$$n \neq 0$$

Prob of False positive

$$\left(1-e^{-km/n}\right)^k \rightarrow$$

optimal $k = \frac{m}{n}(\ln 2)$

Prob in Bloom

$$P = \left(1-\left(1-\frac{1}{n}\right)^{km}\right)^k$$

$n \rightarrow$ size of bit

$m \rightarrow$ items to be filtered.

$k \rightarrow$ hashes.

$$n = -\frac{m \ln P}{(\ln 2)^2}$$

$$k = \frac{n}{m}(\ln 2)$$

FM $\rightarrow$ • in ending r 0's $\rightarrow 2^{-r}$

• m distinct and none of them has tail length of $r = (1-2^{-r})^m$

① $m >> 2^r$

$P = 1 \rightarrow$ finding a tail of length is 1

② $m << 2^r$

$P = 0$ of finding a tail of length.

Scanned with CamScanner

AMS algo

select some no. of variables and assign them random

pos: ~~to~~ elem

$X_1.val = a$ — $X_1.val + (\text{when } a \text{ is})$

$X_2.val = b$ — $X_2.val + (\text{when } b \text{ is})$

$X_3.elem = c$ — $X_3.val \rightarrow$ when c is

Estimation of 2nd Moment

$$\frac{n(2X_i.val - 1)}{\text{no. of var.}}$$

min Surprise

~~if count of each dist becomes $\frac{n}{m}$~~

$S = \left(\frac{n}{m}\right)^2 \cdot m \Rightarrow \frac{n^2}{m} \rightarrow \frac{25}{3} = 8$

Max → count of $m-1$ be 1 and 1 be $(n-m+1)$

$(1^2+1^2+1^2..) + (n-m+1)^2 =$

$m-1$ times

$2+9 = 11$

Reservoir Sampling proof

Base → $n=s$

$P = s/s = 1$

after n elem → $P = s/n$.

after $n+1$ elem

Prob of selecting $n+1^{th}$ + Prob of not sel.

$$\left(\frac{s}{n+1}\right)\left(1-\frac{1}{s}\right) + \left(1-\frac{s}{n+1}\right)$$

$= n/(n+1)$

so $n+1^{th}$ stays.

$$s/n \times \frac{n}{n+1} = s/(n+1)$$

DGIM

uses $\lg^2 N$ bits to represent N bits

No. of buckets $= \lg N$

size of each $= \lg N$

space $= O(\lg^2 N)$

Query Answering in DGIM →

1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	1	1	1	0

20 12 9 6 3 2

when a zero enters → do nothing

but if 1 enters.

- create a size 1 bucket
- if no. of 1 buckets <= 2 ok

else merge 2, 1 size buckets

~~if they form 3, 2-size buckets then merge them and so on.~~

till last bucket

Map Reduce for Matrix

Map

A: (key, val) → $[j, (A, i, a_{ij})]$

B: → $[j, (B, k, b_{jk})]$

Reduce

for each j

Produce $[(i,k), a_{ij} b_{kj}]$

↓

this key, val is fed to another

Map → produces same $(i,k), (a_{ij} b_{jk})$

Reduces

for each (i,k) → sum the values of (i,k)

return that sum.

Scanned with CamScanner

• BFR algorithm

1) discard set →
the clusters are discarded
and their summary is recorded

② Compressed
Points which are close but
not assigned to a cluster
thus are also discarded

③ Retained set
other than these two points
are retained in the hope that
~~a cluster would be formed~~
there

Eg → (5,1), (6,-2), (7,0)
N = 3

SUM = [18, -1]
SUMSQ = [110, 5]

Centroid = $\frac{SUM}{N} = [3, -\frac{1}{3}]$

Variance in d_1 = $\left[\frac{SUMSQ}{N} - \left(\frac{SUM}{N}\right)^2\right]$

$\left(\frac{110}{3} - \left(\frac{18}{3}\right)^2\right), \left(\frac{5}{3} + \left(\frac{-1}{3}\right)^2\right)$

Var = (0.667, 1.56)

Matrix Multiplication in 1 phase
(Computation heavy)

Map → generate
[(i,k), (M, j, m_{ij})] → for M
i → rows of M
k → columns of N
[(i,k), (N, j, n_{jk})] → for N

Reduce
(i,k) → will contain list of values
in the form
(M, j, m_{ij}) or (N, j, n_{jk})

① ~~Sort by j and place first~~
~~the elements of M~~

② ~~Sort by j and place place~~
~~elements of N after the element of M~~

① sort j on M in one list
② sort j on N in one list
for each j
multiply m_{ij} & n_{jk}
from both lists
sum these values
and pair them with (i,k)
((i,k), V) → finally
Sum of all the Vs

Date
Page

$P = \frac{1}{{}^nC_M}$

Size m

$SIM = \begin{cases} \frac{m-1}{m+1} & P = \frac{1}{{}^nC_m}\left({}^mC_{m-1}\right)\left({}^{n-m}C_1\right) \\ \frac{m-2}{m+2} & P = \frac{1}{{}^nC_m}\left({}^mC_{m-2}\right)\left({}^{n-m}C_2\right) \\ \vdots & \\ \frac{m-(m-1)}{m+(m-1)} & P = \frac{1}{{}^nC_M}\left({}^mC_{m-1}\right)\left({}^{n-m}C_{m-1}\right) \end{cases}$

Expected JS_{m-1}
$= \frac{1}{{}^nC_M} \sum_{j=0} \left(\frac{m-j}{m+j}\right)\left({}^mC_{m-j}\right)\left({}^{n-m}C_j\right)$

P. of false pos = $\left(1 - e^{-km/n}\right)^k$

P of bloom Filter = $\left[1 - \left(1 - \frac{1}{n}\right)^{km}\right]^k$

$m = \frac{-m \ln P}{(\ln 2)^2}$

$k = \frac{n}{m} \ln 2$

Scanned with CamScanner

① (3, 6, 9) → 3 6 9 / 7 3 9 / 1 0 9

② (1, 4, 6, 8)

③ (2, 3, 4)

• Reed Solomon :→ (n, k) codes works on m-bit symbols

① Block length, $n = 2^m - 1$ symbols.

② Message size = k symbols

③ Parity check size, $(n-k) = 2t$ symbols

④ Min distance, $d_{min} = 2t+1$

1	1	2	2	3	
1	2	3	4	5	

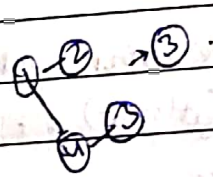
1	1	2	1	3
3	3	3	2	1
1	2	3	4	5

• client receives a webpage from the server, and it will contain some references which will be ~~done~~ rendered by client (fetching from CDN's).

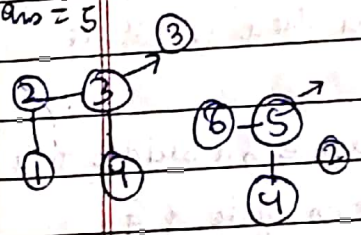
1	5
2	0
3	3
4	
2	

ans = 2
ans = 4
ans = 6
ans = 8
ans = 5

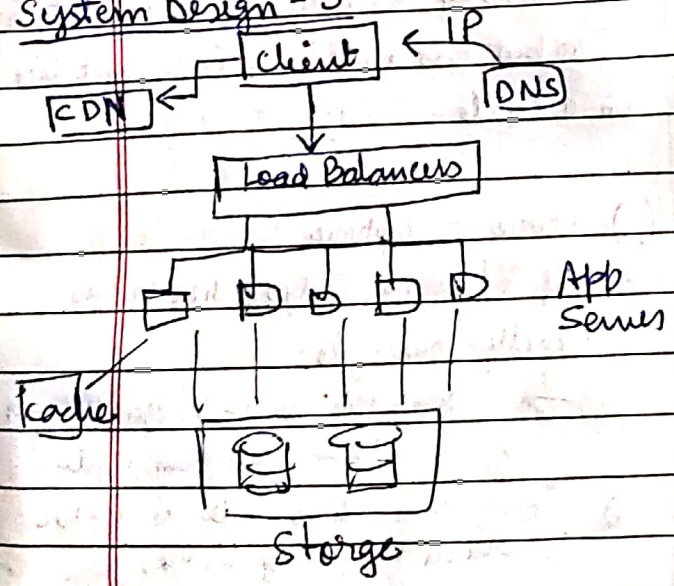
ans = 3
ans = 0
ans = 5



• what would you do if your newsfeed lives in US-East DC but the load balancer sent your to Europe's DC? What can be the workaround.



System Design - 3



Scanned with CamScanner

(Q) For every graph with $\delta(G) \geq 2$, graph contains a cycle.

If no. of edges $>= n$. then the graph G will surely contain a cycle.

n vertices.

$2|E| >= 2n$

$\boxed{|E| >= n}$

Tree → every maximal Path P in a Tree T has 2 leaves as end-points.

T → A graph G is connected if for $\forall$ u,v $\in$ V(G) $\exists$ a u,v path.

(4) A finite graph with $|V(G)|=2$. with odd degree much must contain a path from one to another.

if G is connected → done

If G is disconnected then 2 comp.

H K.

$\sum \deg u = 2E$

↳ no. of odd degree vertices must be even.

⇒ there must exist another vertex in H for the above to happen →

∴ u also $\in$ H →

(5) Let G be a graph where $\delta(G) \geq k$.

(a) Prove G has a path length of at least k.

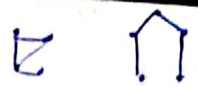
(6) Prove that G is disconnected if $|E| < n-1$.

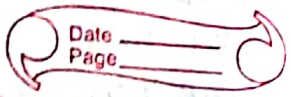
use DSU

there n vertices.

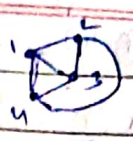
Take an empty set. To merge two vertices we need 1 edge.

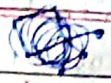
Since G is connected,





k=3





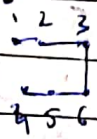
1 2 [1 3 1] 4 .. 124

1 2 1 2 [1 2 1] 3 4

[1 2 1] [2 3 4] k

$|V(G)| = n$

$|E(G)| < n-1$



union operation will perform n-1 times (atleast). ∴ n-1 edges must be there.

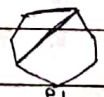
• DSU will also prove that if n edges → then there is a cycle

(Q) Prove that an edge e is a bridge of G if e lies on no cycle of G.

→ Contradiction → e lies on cycle. removing e wouldn't disconnect G. but e is a bridge, i.e. can't lies in a cycle.

(Q) Prove or disprove →

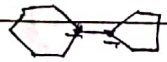
(a) if G has no bridges, then G has exactly one cycle.



X → K_n has no bridges, but can have more than 1 cycle

(b) if a graph with no cycle is a tree which has many bridges

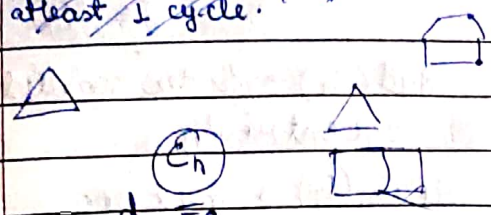
(B) If G has no cut-vertices, then G has no bridges

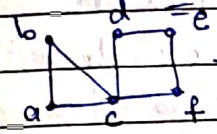


K_2 → has a bridge but no cut-vertex

(6) i) G has no bridges, then G has no cut-vertices.

→ then we can say Graph G has atleast 1 cycle.





→ no bridges but c is cut vertex.

(14) Prove that every 2-connected graph contains at least 1 cycle.

In 2 connected, $\delta(G) \geq 2$.

for this case

G has atleast 1 cycle.

(Q) A graph is bipartite iff it contains no-odd cycles.

(Q) Cayley's Theorem → Any graph with n-vertices has n^{n-2} number of labelled spanning Trees.

- distance from u-v is denoted $d(u,v)$ in G, is the length of shortest u-v path.

→ ① if G is bipartite, then 2 sets X & Y. alternate from X & Y. if there is a cycle, it will be even in length (alt in X & Y)

4400

- let $u \in V(G)$, now partition

$X = d(u,i)$ $i \in V(G)$ = odd

$d(u,i)$ $i \in V(G)$. = even

$X \cap Y = \phi$ & $X \cup Y = V(G)$

- deleting leaves in a Tree in a bottom up manner.

(Q) A tree with atleast 2 vertices has atleast 2 leaves.

n vertices and l leaves.

$\sum_{v \in T} d(v) = 2(n-1)$

$\Rightarrow l + \sum_{\substack{v \in T \\ v>=2}} d(v) = 2(n-1)$

$l + 2(n-l)$

$\Rightarrow 2n - l = 2(n-1)$

$2n - 1 = 2n - 2$

$l = 2$

- A forest with n vertices and k components has exactly (n-k) edges.

- Every graph with n-vertices, m edges contains at least 1 cycle.

consider a ST of T, it must have n vertices but 0 cycles (n-1) edges.

now start adding 1 edge at a time.

each edge will contribute to a cycle.

Rem edges = $m-(n-1) = m-n+1$

This many cycles.

(Q) continue → G is bipartite if it has no odd cycles.

A, B.

→ Let there be an odd cycle

$v_1, v_2, v_3 \ldots v_n v_1$

$v_k = \begin{cases} A, & k \text{ is odd} \\ B, & k \text{ is even} \end{cases}$

now since an odd cycle exists n is odd. v_n and $v_1 \in A$.

but G is bipartite, so, v_n & $v_1 \notin A$

∴ no odd cycles.

$a_1 = \sqrt{p}$ $a_2 =$

4, 8, $a_2 \times a_3$

$a_1 \times a_2$, a_1, a_3, a_1, a_4,

$\Rightarrow a_1^2 = \sqrt{p}$

• if G is bipartite X & Y.
$\Sigma \deg(X) = \Sigma \deg(Y)$

Proof → if we add an edge ~~from~~ then it will contribute 1 to both the vertices.

Say empty graph with 2 sets
1) add 1 edge, $\Sigma \deg(X) = \Sigma \deg(Y)$
2) Suppose it's true for $n-1$ edges.
3) Let's add another vertex.
$\Sigma \deg(X) = \Sigma \deg(Y)$ for all $n \in \mathbb{N}$

(Q) Bipartite is 2-colorable.
X → 1 color, Y → 1 color → positive edges $\exists$ edge (u,v) $u \in X$ and $v \in Y$.

(Q) Brooke's Theorem → Every connected graph G with max degree Δ is Δ-colourable unless the graph is odd-cycle or complete graph.

(Q) If H & G are isomorphic then $\bar{G}$ and $\bar{H}$ are also isomorphic
$G \cong H$ → $\exists f$ st. $V(G) \to V(H)$
$x, y \in E(G)$
$f(x), f(y) \in E(H)$

Equivalently $\exists$ a bijection f
st. ~~$V(\bar{G}) \to V(\bar{H})$~~
$V(G) \to V(\bar{H})$
and if $x, y \notin E(G)$
$f(x), f(y) \notin E(H)$
also, $V(G) = V(\bar{G})$
$V(H) = V(\bar{H})$
means $x, y \in E(\bar{G})$
$f(x), f(y) \in H(\bar{G})$

$d(u,v)$ → shortest u-v path's length

Date
Page

$ecc(v) = \max_{i \in V(G)} \{ d(v,i) \}$

$Rad(G)$ → is the smallest eccentricity

$diam(G)$ → max ecc

center of a graph is the set of vertices with $ecc(v) = Rad(G)$

Periphery → set of vertices whose $ecc(v) = diam(G)$

(Q) For any connected G
$\underbrace{rad(G) \leq diam(G)}_{\text{Trivial}} \leq 2\, rad(G)$

To prove: $diam(G) \leq 2\, rad(G)$

let c be a central vertex
$d(u,c) \leq rad(G)$
$d(v,c) \leq rad(G)$
$\Rightarrow$ by Δ in equality
$d(u,v) \leq 2\, rad(G)$
$\downarrow diam(G)$ = assumed at the beginning

(Q) if $\delta(G) \geq k$, then G has a path length of at least k.
Since $\deg(v_0) \geq k$.
v_1 → can be chosen $k-1$ ways.
$v_0 v_1$ → v_2 can be chosen in $k-2$ ways
~~v_j can be~~ if v_j is chosen then $k-j$ ways to choose v_{j+1}

Scanned with CamScanner

(Q) if x is periphery of G and $d(x,y) = ecc(x)$ then prove y is also in periphery.

$ecc(x) = diam(G)$

$d(x,y) = diam(G)$

$ecc(y) \geq diam(G)$

but $diam(G)$ is max

$\Rightarrow ecc \leq diam(G)$

$\rightarrow$ y is in periphery

(Q) if u & v are adjacent their ecc would differ by at most 1.

$\rightarrow$ u_1 st. $d(u,u_1) = ecc(u)$

v_1 st. $d(v,v_1) = ecc(v)$

we know,

~~$d(u,u_1) \leq d(v,v_1)+1$~~

$d(u,u_1) \leq d(v,u_1)+1$

and $d(v,v_1) \leq d(u,v_1)+1$

↳ because u & v are adjacent

$ecc(u) \leq d(v,u_1)+1 \leq ecc(v)+1$

$ecc(v) \leq d(u,v_1)+1 \leq ecc(u)+1$

(Q) 1–n are arranged in some order around a circle.

sum of k-adjacent taken k at a time

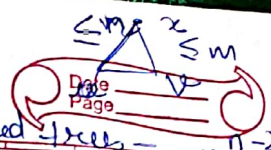
$S = \frac{k\,n(n+1)}{2}$ $\quad$ 11 22 33 44 $\quad$ kn

↓

Total pigeon

n holes

One hole with $\frac{k(n+1)}{2}$



no. of labelled trees $= n^{n-2}$

(Q) Suppose that u & v are vertices in a graph G, $ecc(u) = m$, $ecc(v) = n$.

$m < n$, Prove $d(u,v) \geq n-m$

let $x \in V(G)$

$d(u,x) \leq \cancel{ecc(u)}\ m$

$d(v,x) \leq \cancel{ecc(v)}\ n$

$d(u,v) \leq \cancel{d(u,x)+d(v,x)}$ $m+n$.

suppose

$d(u,v) > n-m$

(Q) suppose that a connected G is decomposed into G_1 and G.

then they have common vertex

$V(G_1) \cup V(G_2) = V(G)$

$E(G_1) \cup E(G_2) = E(G)$

$E(G_1) \cap E(G_2) \cap \phi$

But to show $V(G_1) \cap V(G_2) \neq 0$

suppose $= \phi$

$\nexists$ u,v such $u \in V(G_1)$ and $v \in V(G_2)$

$u,v \in E(G)$

$\exists (x,y)$ st. no path

↓ disconnected → contradiction

(Q) a connected G with even degree has no bridges.

contradiction

say a bridge is there.

Two comp.

H & K.

H has $\Sigma deg = 2E$

each vertex will have even degree

K, $K-1$

$k+k$ ... odd $\neq$ even

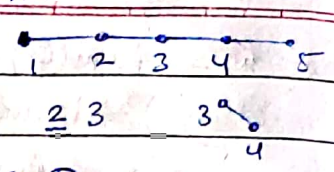
T→ Vertex

A Graph G is k-connected iff every pair of vertices are joined by k-pair wise disjoint paths.

$\frac{{}^{14}C_7}{8} = \frac{14 \times 13 \times 12 \times 11 \times 10 \times 9 \times 8}{7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 8}$

$13 \times 7 \times 11 \times 3$

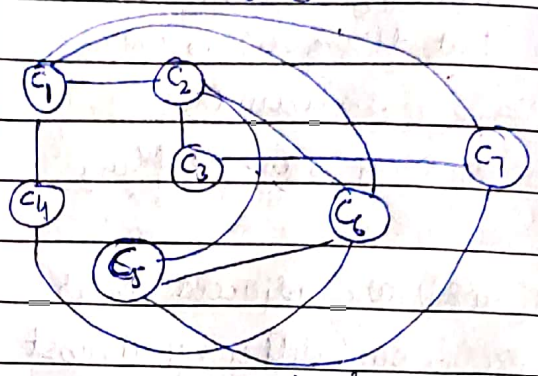
Date _____
Page _____



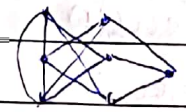
(a) if G is connected and diam(G) ≥ 3 then $\bar{G}$ is connected

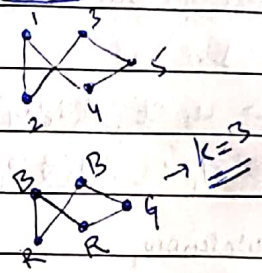
(b) Prove that if diam(G) ≥ 3, then diam($\bar{G}$) ≤ 3

(c) Prove that if G is regular and diam(G) = 3, then diam($\bar{G}$) = 2.



(a) Prove that G is a Tree if ∀ u,v ∃ only one path b/w them



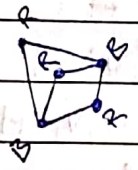


Prim's alg

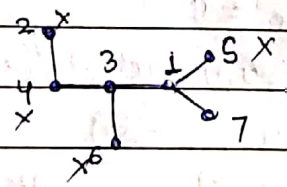
① choose a v, mark it.

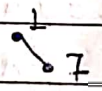
② From all the edges that have one marked vertex find the min and mark the edge.

③ stop when all vertices are marked.



Find Prufer





~~2~~ → 4

~~4~~ → 3

~~5~~ → 1

~~6~~ → 3

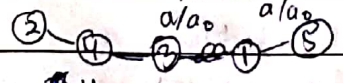
~~3~~ → 1

4 3 1 3 1

G get the tree back from this.

1, ~~2~~, 3, 5, 6, 7

$S_0 = 2$ → S/S_0 a/a_0 S/S_0 a/a_0



$S_0 = 4$

① if G is connected and diam ≥ 3 then $\bar{G}$ is connected.

We'll try to prove the contrapositive

suppose $\bar{G}$ is disconned, then we shall prove diam(G) ≤ 2

→ let $C_1, C_2 .. C_n$ be components of $\bar{G}$

• In G → there is an edge b/w any 2 vertices that belong to diff comp. of $\bar{G}$

• For any two x, y belonging to comp of $\bar{G}$, there is a path connecting them via a vertex belonging to a diff connecting comp.

⇓ diam(G) ≤ 2.

diam(G) = 1 occurs when G is empty.

Scanned with CamScanner

(b) Let $diam(G) \geq 3$
if G is disconnected
then $\bar{G}$ is connected with $diam(\bar{G}) = 2$

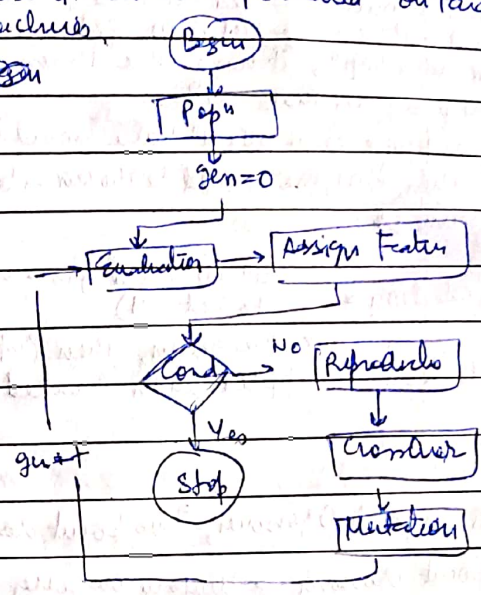
(e) bridge iff inside every ST.

① Let T be ST and e be a bridge not in T.
$G \setminus \{e\}$ is a disconnected graph.
which also makes T disconnected
G not true, ∴ all bridges must be in a ST.

② suppose e is not a bridge
$G \setminus \{e\}$ is connected, $V(G) = V(G')$
T is a ST of G doesn't contain e. Therefore any edge in ST is a bridge.

• Advantages of Binary Coded
• They work with coding of decision variables not decision variables themselves. Search space is always discrete

• GA work with a popⁿ instead of a single solⁿ.
• GA dont require gradient information.
• GA use probablistic rules to guide the search.
• These GA can be implemented on Parallel machines.



EC

• Binary Encoding

$$x_i = x_i^{min} + \frac{x_i^{max} - x_i^{min}}{2^{l_i} - 1} \quad \forall i$$

• Features of Binary Type Encoding
① arbitary (also exit finite precision) can be achieved
② by using different lengths, different precision can be found.
③ Decision variables are allowed to take + and −

• Difficulties with classical and Gradient Methods →
① Convergence depends on Initial solⁿ
② Most algo either get stuck on suboptimal solⁿ
3. One algorithm may not be fit for every problem.
④ Algorithms can't be parallelized

• Problems in Real World
① search space contains more than 1 optimal solⁿ of which algorithm are attracted to local ones.
② Some variables are restricted to discrete values
So during optimization assume them to be continuous
Post optimisation lower and upper bound needs to be checked separately → 2^n operations

• Types of Selection of
① Tournament ② Proportionate
③ SRWS ④ Ranking

• Tournament → each solⁿ is allowed to take part in 2 tournaments. Best = 2 worst = 0
• Roulette Wheel
Wheel is divided into N-sections, where each is f_i / f_{avg} → wheel is spun N times, choosing the solⁿ indicated by the pointer.
• P of selecting ith → $f_i / \sum f_i$
• O(N) solⁿ are generated

• Proportionate Selection
ith var will contain = Index till the point where pointer remained true.
Pointer = total f × rand()
then keep subtracting fitness values till pointer is true.

• SRWS
• The p_i are multiplied by Popⁿ size and the expected no. of copies is calculated for all the solⁿ.
Next RW is perform.
$p_i N$ is calculated

- Stochastic universal sampling (SUS)

① A random, r is chosen. number are the N equi spaced.

gen → $R = \left(r, r+\frac{1}{N}, r+\frac{2}{N} \ldots r+\frac{N-1}{N}\right) \% 1$

- So choose r, then other sol^n will be found by traversing $\frac{1}{N}$ along the periphery.

Issues with Proportionate Selection

- The outcome is based on true fitness and not on the relative fitness. So, if in a pop^n, it has a large fitness value the $p = 1$ for that sol^n.
- if suppose even all sol^n have around same p then everyone would be chosen → leading to same sol

Ranking selection → • Sol^n are sorted from worst (1) to best (N)

- Ranks are given, then Proportionate selection is applied with ranked fitness.

- Crossover
  - Single point Crossover, • Two point crossover
  - k-point crossover • Uniform crossover (Each bit is chosen from either parent. then mixing ratios are used)

- If crossover prob. is P_c then $100 P_c$% strings in the pop^n are used in crossover and $100(1-P_c)$% are simply copied

Mutation

- Bit flip operator changes 1 → 0, or 1 with P_m prob.
- Mutation clock operator

choose a random r, then estimate the next bit after skipping $= -P_m \ln(1-r)$ bits

$O(1/P_m)$ less random no. are generated

- Elitism → the preservation and use of previously found best sol^n in subsequent gen.
  - Causes an overall increase in performance
  - No degradation in gen.

- Elite preserving operator

best ε% are copied

$(100-\varepsilon)$% go to preserving

- Create offspring of current population then choose the best N sol^n out of 2N

Constraint Handling

① Preserving feasibility

Explicit (reduces no. of variables)

$h(x) = 2x_1 - x_2^2 x_3$

$x_1 \to 0.5 x_2^2 x_3$

For k linear simultaneous equations, k decision variables can be eliminated only (n-k) decision variables need to be used

- Implicit

$h(x_k, (x \setminus x_k)) = 0$

- uses all n decision variables
- but while finding root x_k is not taken into consideration

2 ways to handle x_k

- Lamarckian → substitute
- Baldwian → don't substitute

② Penalty F^n

$F(x) = f(x) + \sum R_j \langle g_j(x) \rangle + \sum r_k |h_k(x)|$

↳ all the constraint must be normalised, (because of different order of magnitude)

now $F(x) = f(x) + R\left[\sum \langle g_j(x) \rangle + \sum |h_k(x)|\right]$

using fixed R is called as static penalty method.

- Issues with static penalty
  - different values of R is tested → Exhaustive
  - R distorts the objective f^n

Dynamic Penalty

$F(x) = f(x) + \underbrace{(C \cdot t)^\alpha}_{\text{dynamically changes for different } t} \left[\sum_j \langle g_j(x) \rangle^\beta + \sum_k h_k(x)^\beta\right]$

- Methods based on feasibility.

Behavioral Memory Method →

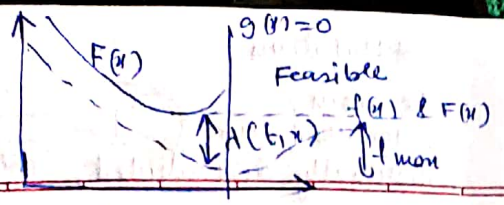
All constraints are considered in a sequence. When sufficient feasible sol^n satisfying the constraints taken from the top

Powell and Skolnick

$$F(x) = \begin{cases} f(x) + R\left[\sum_j \langle g_j(x) \rangle + \sum_k h_k(x)\right] + \lambda(t, x) & \text{if } x \text{ is infeasible} \\ f(x) & \text{otherwise} \end{cases}$$

$\lambda(t, x)$ → difference b/w best static penalized f^n and worst feasible sol^n.

Scanned with CamScanner



Blend Crossover → BLX-α

$P_1 = x_i^{(1,t)}$

$P_2 = x_i^{(2,t)}$

Date
Page

BLX-α chooses randomly from.

$$\left[x_i^{(1,t)} - \alpha\left(x_i^{(2t)} - x_i^{(1t)}\right),\ x_i^{2t} + \alpha\left(x_i^{(2t)} - x_i^{1t}\right)\right]$$

if u_i is random b/w (0,1).

$$x_i^{(1,t+1)} = (1-\gamma_i)\, x_i^{(1t)} + \gamma_i x_i^{(2t)}$$

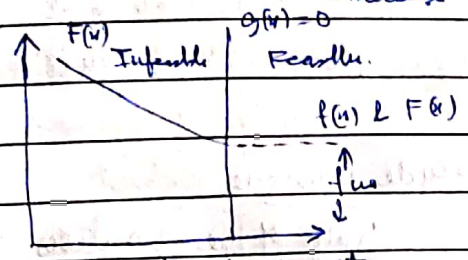
$$\gamma_i = (1+2\alpha)\, u_i - \alpha.$$

- If $\alpha = 0$, → random sol^n in $(x_i^{(1t)}, x_i^{2t})$
- BLX-0.5 is better than any other α.
- location of offspring depends on

$$x_i^{(1,t+1)} - x_i^{(1,t)} = \gamma_i\left(x_i^{(2t)} - x_i^{(1t)}\right)$$

Modification by deb →

$$F(x) = \begin{cases} f_{max} + \sum g_j(x) + \sum h_k(x) & \text{if } x \text{ is infeasible} \\ f(x) & \text{otherwise} \end{cases}$$



- Here the objective f^n value is not computed for infeasible sol^n
- 0 constraint violation for feasible

Tournament selection happens →

① Feasible > infeasible

② For 2 feasible, feasible with > better obj.

③ 2 infeasible with smaller constraint violation

SBX → Simulated Binary

$u_i \in [0,1]$ → random. n_c → user defined

Calculate βq_i: $\quad u_i \le 0.5$

$$\beta q_i = \begin{cases} (2u_i)^{1/(n_c+1)} & \\ \left[\dfrac{1}{2(1-u_i)}\right]^{1/(n_c+1)} & \text{else} \end{cases}$$

~~then offspring are~~ →

$$x_i^{(1,t+1)} = 0.5\left[(1+\beta q_i)\, x_i^{(1t)} + (1-\beta q_i)\, x_i^{2t}\right]$$

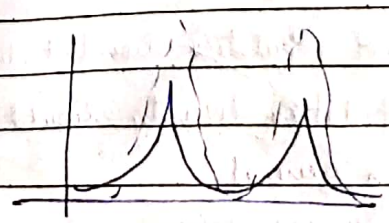
$$x_i^{(2,t+1)} = 0.5\left[(1-\beta q_i)\, x_i^{1t} + (1+\beta q_i)\, x_i^{2t}\right]$$

- It has similar search power as one-point crossover.

$$p(\beta_i) = \begin{cases} 0.5(n_c+1)\,\beta_i^{n_c} & \beta_i \le 1 \\ 0.5(n_c+1)\dfrac{1}{\beta_i^{(n_c+2)}} & \text{else} \end{cases}$$

β_i is ratio b/w differences among ~~offspring and parent~~

$$\beta_i = \left[x_i^{(2,t+1)} - x_i^{(1,t+1)}\right] \Big/ \left[x_i^{2t} - x_i^{1t}\right]$$



Both SBX and BLX can be written as

$$x_i^{(1,t+1)} = (1-\gamma_i)\, x_i^{(1,t)} + \gamma_i x_i^{(2t)}$$

BLX → $\gamma_i = (1+2\alpha)\, u_i - \alpha$

SBX → $\gamma_i = 0.5(1 \mp \beta q_i)$

Method based Decoder

- chromosome stores information about how to fix an infeasible sol^n.
- decoder → mapping b/w feasible sol^n and a decoded sol^n.

Crossover

→ Linear, → Naive, → Blend Crossover,

→ Simulated Binary, Fuzzy Recombination,

Unimodal Normally, Simplex, Fuzzy connective

Unfair usage

① Linear Crossover → Requires two parents

$x_i^{(1,t)}$, $x_i^{(2,t)}$

now generate offspring as you like

② Naive $P_1 = \{x_1^{(1,t)}, x_2^{(1,t)} \ldots x_n^{(1,t)}\}$

$P_2 = \{x_1^{(2,t)}, x_2^{(2,t)}, \ldots x_n^{(2,t)}\}$

$OS_1 = \{x_1^{(1,t)}, x_2^{(1,t)}, x_3^{(1,t)}, x_4^{(2,t)} \ldots x_n^{(2,t)}\}$

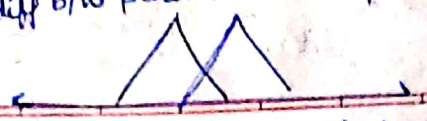
- $OS_2 = \{x_1^{(2,t)}, x_2^{(2,t)} \ldots x_4^{(1,t)} \ldots x_n^{(1,t)}\}$

↳ same as one point crossover

Scanned with CamScanner

Fuzzy Recomb op

• diff b/w parents are computed.



gives importance to sol^n closer to parents

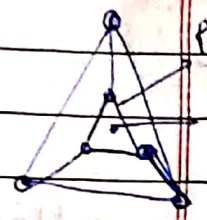
$d(x_i^{2t} - x_i^{1t})$

if d is small, near sol^n are generated

Simplex Crossover →

• More than 2 parents participate in the crossover.

(n+1) parents where n decision variables.

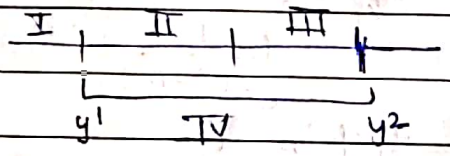


Parent Simplex.

centroid

~~First centroid~~ is calculated

Fuzzy Connectives → For each 4 regions are marked.



Unfair Avg Crossover.

favour the better parent for offspring gen.

$$x_i^{(1,t+1)} = \begin{cases} (1+\alpha)\, x_i^{(1,t)} - \alpha x_i^{(2,t)} & i=1,j \\ -\alpha x_i^{(1,t)} + (1+\alpha)\, x_i^{2t} & i=j+1,n \end{cases}$$

$$x_i^{(2,t+1)} = \begin{cases} (1-\alpha)\, x_i^{(1,t)} + \alpha x_i^{(2,t)} & i=1,j \\ \alpha x_i^{(1,t)} + (1-\alpha)\, x_i^{(2,t)} & i=j+1,n \end{cases}$$

• Assumption

① mean of Population shouldn't change.

② The population diversity should increase in general.

↓ Fuzzy connective and unfair average don't preserve population mean.

Mutation

Random,
Non-Uniform,
Normally distributed, Polynomial

• PMX → Partially Mapped X

(123|456|78) (375|168|24)

4-1, 5-6, 6-8

(...|168|...) (...|456|...)

• Cycle crossover

half of the positions are preserves. (preserve in pos 1,2)

rest all is copied from random parent

(12345678) (24687531)

1 2 • 4 • • • 8

1 2 6 4 7 5 3 8

Order Crossover (OX1)

(12|345|678) (24|687|531)

(87|345|126) (345|687|12)

order from 2nd parent

~~start from 2nd~~

• OX2

select some random position from 2nd parent

12345678 24687531

12346578

↳ order in 2nd parent

• Position based Crossover

(1 2 3 4 5 6 7 8) (2 4 6 8 7 5 3 1)

put them in same pos in P2

→ 1 4 6 2 3 5 7 8

↳ then put each element from P_1 itself

• VR (Voting Recombination)

p-sexual crossover

τ → threshold.

p=4 τ=3 1,2,6 > τ.

1 4 3 5 2 6

1 2 4 3 5 6

3 2 1 5 4 6

1 2 3 4 5 6

1 2 4 5 3 6

↳ any permutation of remains

Alternating Pos ~~cro~~ X

selects alternate elements from P_1 and P_2.

(1 2 3 4 5 6 7 8) (3 7 5 1 6 8 2 4)

(1 3 2 7 5 4 6 8) (3 1 7 2 5 4 ...

Displacement Mutation operator (DM)

a substring is chosen and put some where else

1 2 3 4 5 6 7 8

→ 1 2 7 3 4 5 6 8

• Exchange Mutation → swaps two random positions

• Insertion Mutation

random chooses one and inserts it at some other random position

• simple inversion

choose a substring and reverse it

• Inversion Mutation ope

call simple Inversion

call displacement

• Scramble

choose a substring and shuffle the elements

Convex

if for any 2 pair of sol^n.

$f(\lambda x^1 + (1-\lambda) x^2) \leq \lambda f(x^1) + (1-\lambda) f(x^2)$

↳ local min is alway global Min

↳ Hessian = $\frac{\partial^2 f}{\partial x_i \partial x_j}$

• ideal objective Vector ↦ lower bound of each obj

it corresponds to a non-existent sol^n.

The only way an ideal objective vector corresponds to a feasible solution is when the minimal sol^n of all obj are identical

utopian objective

marginally smaller than ideal

• Nadir objective

contains upper bound of objectives present in entire Pareto optimal set.

$f_i^{norm} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*}$

• Weak Pareto
• Strong Pareto } → (defined in the non-dominated set)

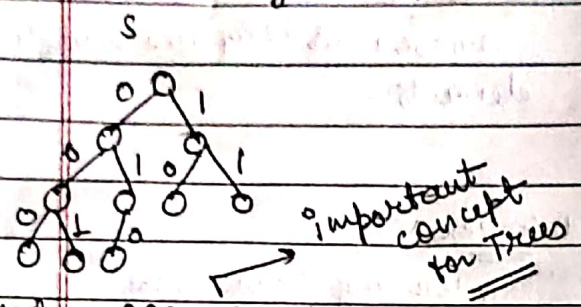
Scanned with CamScanner

(Q) Find total no. of nodes in a complete Binary Tree.

① O(n) approach (Bad)

② $K \log n \rightarrow$ Binary search.

Time for ~~binary~~ valid f^n



Leaf Nodes → 000, 001, 010, ∞
0 1 2.

Write a f^n isNodePresent(i)
↳ find the rightmost i

then Total nodes will be
$2^{h-1} + i$

② Variant of Djikstra.

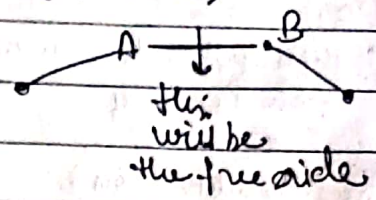
- you have a coupon and you can travel from one city to other for free.

1) Find all paths. (For each path remove max edge and track the answer).

② one by one make edges 0 and apply djikstra.
(E) (E lg(V))
→ $E^2 \lg V$

③ Apply 2 djikstra from S→D and D→S.

for all edges
A — B
↓ ↓
SD(A) DS(B)

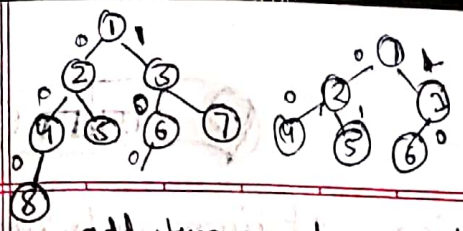


C — D
SD(C) + DS(D)

find min among these edges

(Q) given X and Range [A, B]
find $X \oplus y$ where $y \in [A\ B]$
such that $X \oplus y$ is max.
→ start with the MSB bit
put the toggled bit in the generating num. If the number is less than B and greater than A → then fine else put the same bit.

(Q) Rain Water → 2 (3D Version)
put Boundary cells in min heap →
We are starting from end nodes and checking for the contribution if the cell is ~~less~~ having lesser height



• add two numbers as ll.

3→2→1

4→9→9→2

1→2→3→4

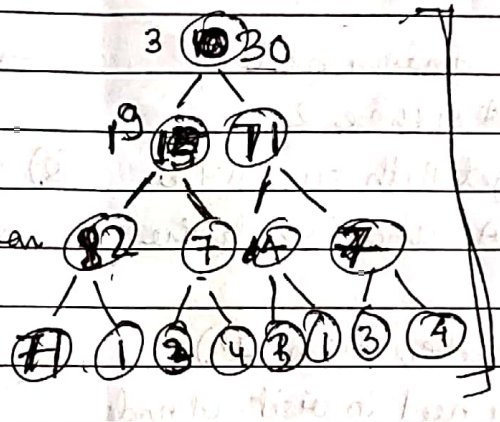
2→1→

after 10 iteration

cost after 10 it = 7 → near

cost after 10 it ≈ 0

change = $\frac{7-0}{7}$



• 1st meaning +ve

guess 0 12 → 3 | 124 → 7

actual 123 → 6 | 123 → 6

$\frac{3\times4}{2} \Rightarrow 6$

12 4 | 2, 10, 5 | 123

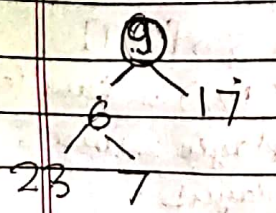
123 | 123

698 | 894

745 | 765

1 2 3 4 5 6 7

i-B,



(-1,0) , (4,3)

(-1,2,3) f = -1

(23,-4) f = -4

-9,4,8,-1 f = -9

1 2 3 4 5 | $\frac{n(n+1)}{2}$

1 0 1 0 1 | $^3C_2$

$\frac{2\times1}{2} = 1$ +

5 17 100 11 | 0 3

5 17 11 | 2

1 2 3 4 5

1 0 1 0 1

~~#TINY URL~~

#MISC - 6

(Q) Given a BT, find max path sum.

① construct an undirected graph and then run Floyd Warshall on top of it.

② choose 2 random nodes
→ get LCA of those 2.
→ then get Path sum via the LCA and track the max sum.

• you can't do better than $O(n)$ because you need to visit all nodes atleast once.

lsum
rsum

only ~~take~~ +ve

cur_sum = max(lsum, 0) + max(rsum, 0) + val.

if (cur_sum > max)
max = cur_sum.

return val + max(left, r, 0)

(Q) Majority element > n/2

count == 0
cur = a[i]
count = 1
else if a[i] == cur
count++

intuition → cancel if someone better comes out

(Q) Given 2 strings with backspace key → mean abc\b ⇒ ab
abc\b\b ⇒ a

string length is too big (it can't be stored in mem)
→ just use a stack.

(Q) given an array, find bit-wise OR of all the subarrays.

for each xth bit (0-31)
↳ find no. of subarrays having x^{th} bit set.

for n elements

x^{th} bit comes out → 0 0 0 1 1 1 0 1 0 1

$2^i \left[\frac{n(n+1)}{2} - \text{subarray formed by consecutive 0's}\right]$

(Q) Find Max. Fruit Packets →

A B C → exactly 3
atleast 1 A, atleast 1 B.

Find max Fruit Packets

$O(1)$ $\left(\min(B, R), \frac{B+R+W}{3}\right)$

(Q) Given N digits → N → even
some digits are denoted by '?'

Turn → Replace ? → [0-9]

now if sum(N/2) = sum(end N/2)
↳ player 2 wins.
else player 1.

$\frac{L+R}{2} = 0$ → P_2 wins
else P_1 wins.

L = min Balance
R = max Balance

Bal = sum L − sum R

Scanned with CamScanner

backspace
⇒ ab
⇒ aB
big
is menj)

ind bit-wise
ways.
0-31)
ways

$\frac{N}{2}$

1 2×3
0 1 1 1 0 1 0 0

secutive 0's]

→

atleast 1 B.

+w)

rd by '?'

(end N/2)

(Q2) Consecutive Numbers

q → 4+5, 2+3+4, 9

3 ways

• elements should consecutive

$N = (x+1) + (x+2) \dots (x+k)$

$N = kx + \frac{k(k+1)}{2}$

$x = \frac{\left(\frac{2N}{k} - k - 1\right)}{2}$

each integral x will contribute to soln.

upper limit of k. $O(\sqrt{N})$

more precise $k^2 <= 8N$

or no. of factors of N

#Flipping columns

q → 2 zero → 2 column.

$k \% 2$

$k <= x$

① → if $xy.k == 0$

k zero.

else no

1 → 1, 2

2 ↓ → 4, 7, 1

↑ → 1, 5, 6

3 → : 1, 6, 7

4 → L, ↓ → 1, 3, 6, 7

⑤ Memory leaks

• due to not deleting the allocated memory

• avoided by using smart pointer

• deleting the allocated memory carefully.

1, 2, 3, 4, 5, 6

Date
Page

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

(0,0) (1,2) (2,5) (6,8) (100,100) (120,120)

(0,0) 0 3 7 14
(1,2)
(2,5)
(6,8)
(100,100)
(120,120)
(120,100)
(100,180)

1 2 3 4 5 6
0 0 0 1 0 0
1 2 3 4
2+3+4+4 = 13

• Kahn's Algo for Top Sort →

① push all nodes with in-deg = 0

② decrement degree of neighbours by 1
if degree == 0 push these nodes to q

③ after removing vertex, nodes++

• obs → if count of nodes != total nodes.
then no Top Sort possible ⇒ cycle in the graph.

② DS used to manage files and folders in Mobile.

Trees → modified nodes.
node → Name.
List Directories
List Files

③ Code for Djikstra → n^2 / $n \log n$

④ Find k^{th}-min element in BST

Scanned with CamScanner

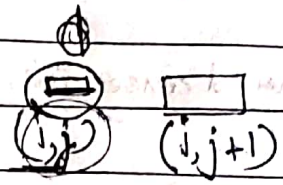
- Remove Loop from LL.
  find the loop Node
  now for every node,
  check

- Find the max and 2nd max in an array →
  1. Sorting
  2. Two traversals.
  3. update both max and min at the same time

- what are disadvantages of a Stack
  - limited size based on OS
  - always memory is assigned contiguously.

- How can you show Heap overflow by a program →
  1. keep allocating ~~too~~ space to large no. of variables.
     10^6 → ints are allocated
  2. allocate a very big size
     si

stack overflow
1. mat[10^5][10^5]
   ↳ stack overflow
2. if a f^n recursively calls itself infinite times → then also stack overflow occurs.



→ 1, 3, 6, 7

(i,j) (i,j+1)

right side → left

1, 3, 4, 5 | 1, 3, 6, 7

(i,j+1)

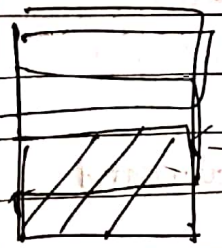
up | down

1, 2, 4, 7 | 1, 2, 5, 6.

(i,j) | (i-1,j)

(i,j) 1

dp[i][j] = 0
1 + max(dp[i-1][j], dp[i][j-1], dp[i][j+1])



dp. [5]

Scanned with CamScanner

Scanned with CamScanner

(0(5 (6 ( ) ( )))

Date
Page

solve (dest, int cost, int & ans, prev —, ~~int~~

~~if & cost~~

class

Class Class

Class

4 3 1 2 2 2

12
6
4
3

18

• Morris Traversal

AA

10

1 2 3 4 5 6 7 8 9 10

1010

1+(1+1)+(2+1)+(1+1)+(3+5)+(1)+(1+1)+ 1+1+ 2+1

1+2+3+2+3+1+2+2+3

(Q) What is Page Fault and why it occurs?

(Q) What is Virtual Memory?
a storage allocation scheme in which Secondary Memory can addressed as though it were part of main memory.

(Q) What is thrashing?
When high degree of Multiprogramming occurs, then less no. of frames to each process → CPU keeps on swapping pages from

• circular Queue

(Q) How do you recover from Thrashing?
→ Keep a threshold that after a certain no. of processes → no further process will be serviced.
→ if already in thrashing, then suspend ~~or kill some~~ processes

• Network Layer
• Burst Balloon – DP
• Dijkstra (N/gn)
• Virtual Fn, VTA
• Kruskal, Prim
• Bellman, Floyd Warshall

Scanned with CamScanner

• BellManFord

① Relax the edges V-1 times

for i → 0 to V

for j → 0 to E

u = E(i).src

v = E(i).dest

w = E(i).weight

if dis[v] > dis[u] + wt

dis[v] = dis(u) + wt

now check for negative cycle.

for i = 0 to E

if dis[u] + wt < dis[v]

cycle

- we relax the edges V-1 times because the idea is after ith iteration nodes having at most i edges has been calculated

• Floyd Warshall

The idea of Floyd Warshall is if K is not an intermediate vertex b/w i and j then dist(i)(j) remains same

else dist(i)(j) = dist(i)(k) + d(k)(j)

if d(i)(j) > d(i)(k) + d(k)(j)

dist is same as given i/p graph

• ESOP Tables

1. Users → UserName, Type, Pass, Email
2. Customer → Type of Service, SID.
3. Seller →
4. History Table → amount, Product.
5. Service → SID, Service Id, Seller ID.
6. Restaurant → RestName, ServiceID, item1, item2, item3
7. barber →
8. E-rickshaw →

Why → ~~when~~ that time it was a need but not serviceable.

Challenges Faced

① Payment Gateway (Not Solved)

②. Handling User Cart (cookies, then dB table)

• Your Role

FrontEnd → Login, Register, Restro Ajax.

Backend → Endpoints, Payment Api, Profile, Fn of Restaurant, Usercart

• LSTM

Action Recognition Dataset KTH

• If not LSTM, then we'll be ignoring the Time Series information only spatial Features will be captured

Variants with LSTM →

① PCA → reduced Features

② Encoders → reduced Features

- B^{th} smallest Prime Fraction

1, some prime numbers.
push all $1/a(i)$ $1 \le i \le n$ into min heap.
then pop and push the new numerator keeping the denominator const.

$O(k \lg k)$

100, 101, 102, 103, 104, 105

Date
Page

$100 + 105$
$\Rightarrow 205$

$6C_3 \rightarrow \frac{6 \times 5 \times 4}{3 \times 2} \Rightarrow 40$

- Strobogrammatic number

(3, 5)

(1, 5) → (1, 0, 8)

010	1	0
818	1 - 2 ✓	1 - 3 ×
111	2 - 3	2 - 1
916	3 - 1	3 - 2

$4C_1$

$6C_3 \rightarrow \frac{6 \times 5 \times 4}{3 \times 2} = 40$

$1 \rightarrow 2$
$2 \rightarrow 2$
$3 \rightarrow 2$ $\rightarrow 3C_1\ 2C_1$
$\Rightarrow 6$

$6C_3 \rightarrow \frac{6 \times 5 \times 4}{3 \times 2} \Rightarrow 40$

- double trouble.

5	1	0
35	0	1 - 2
(5555555)		2 - 1
(10, 55555)	2	

2222

$4C_3 \Rightarrow 4$

123123

1231
123
231
131
121

101
011
110

1001
001
100

$My = Nx^2 + 0x + P.$

(A, B)
(x, 0)
(x, y)

$(A-x)^2 + (B-y)^2 = x^2 + y^2$
$A^2 + x^2 - 2Ax + B^2 + y^2 - 2By = x^2 + y^2$
$A^2 + B^2 = 2Ax + 2By$

$d_1 = (A-x)^2 + (B-y)^2$
$d_2 = (x_1 - x)^2 + y^2$

$d_1 = d_2$ and (x, y) on given curve.

$(A-x)^2 + (B-y)^2 = (x_1 - x)^2 + y^2$
$A^2 + x^2 - 2Ax + B^2 + y^2 - 2By = x_1^2 + x^2 - 2x_1 x + y^2$
$A^2 + B^2 - x_1^2 = 2Ax + 2By + 2x_1 x$
$A^2 + B^2 - x_1^2 = (2A + 2x_1)x + 2By$
$(A, B) = (1, 1)$ $x_1 = 1$
$2 - 1 = 4x + 2y$
$\boxed{4x + 2y = 1}$

$My = Nx^2$
$Nx^2 - My + 0x + P = 0$

110
101
011

1010
0110
$\boxed{1100}$

Scanned with CamScanner

(x,y)

(A,B) $(x_1,0)$

$(x-A)^2 + (y-B)^2 = (x-x_1)^2 + y^2$

$x^2+A^2-2xA+y^2+B^2-2yB = x^2+x_1^2-2xx_1+y^2$

② h → ~~1~~, ~~2~~, 3, 1

D → ~~3~~

1→

2→

3→

(Q) Longest Consecutive Sequence

- add elements to an unordered set.
- then for each element if $a(i)-1$ is not present, then this would be the start of the seq. continue counting the elements.
- keep track of max elements.

3 How

2

Horse digharat c=0

~~2~~, 3 ~~3~~ ~~1~~ → 1 0

Her di c=0

1 2 c=1

2 1

H D

1,2 2

(Q) Longest increasing consecutive seq.

dp → $O(n^2)$ ✓ (using LIS)

dp → $O(n)$ → using consec prop

if $A(i)-1$ is present

$dp(i) = dp(ind)+1$ then $A(i)$ will add 1 to that prev seq.

else it will be start of the seq. $dp(i)=1$

1 → 2

3

1 → 3

2

1 → 2

dp, Graph, LL,

8957528473

1 2 1, -2

1	2	
1		

1 2 3

Proj → Wachovia

~~(30, 1), (3, 8)~~

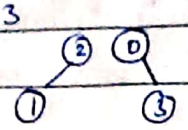
(178, 12), (30, 1), (13, 7), (34, 8), (37, 6)

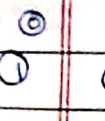
← Couple Holding Hands →

0 1 2 3

2, 3, 1, 3

~~(2 = 1)~~





In the Github Repo → do the proper documentation of codes. (Very Imp)

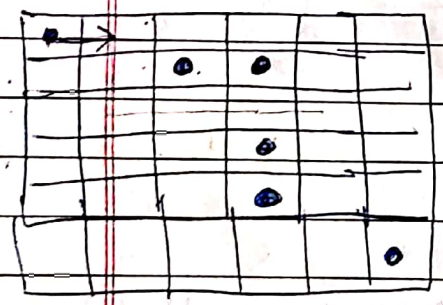
GSOC

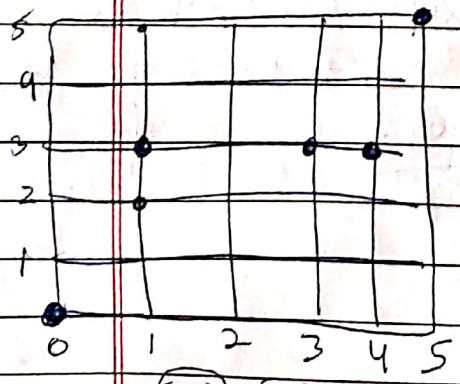
① AIMA Code → 1. Go through the Algo. 2. Read the Code Base and Mark which which are yet to be done ③. Do them.

② Checker Frame Work → ① Read the Code Base.

③ Cloud CV → ① Go through the code Base and get the picture.

② Collecting Packages →





(1, 2), (1, 3)

(3, 3) → collect vertically

(4, 3) → maintain lists over each x-coordi and collect all packets vertically.

(5, 5)

CF - Round 615 - div 3.

① Collecting Coins →

Alice - a coins Carene - c coins

Bob - b coins

Poly → n coins

$A = n - (B + C)$

$a + n - (B + C)$

$b + B$

$c + C$

$a + n - B - C = b + B = c + C$

$a + n = b + 2B + c$

$a + n = c + 2c + B$

m coins 5 15

$r_a = m - a$ (10)

$r_b = m - b$ $r_a = 5$, $r_b = 7$, $r_c = 8$ ⇒ $20 \geq c$ go left

$r_c = m - c$

5 9 (7)

$r_a = 2$, $r_b = 4$, $r_c = 5$ → ≥ 8 go left

(2, 3)

5 6 → (5) (3, 4) (3, 2) (4, 1)

$r_a = 0$, $r_b = 2$, $r_c = 3$ → $5 \leq 8$ → go right

$r_a = 1$

$r_b = 3$ (8) → Yes

$r_c = 4$

← Binary Search on the target Value →

if reaches exactly → Yes

else → No

Scanned with CamScanner

③ Product of 3 numbers

① Find Factors
↳ order → 10^4.

Java Flower Shop

Date
Page

out of these we need 3 integers.
sort these factors.

minimize a and maximize c.

a → min factor of n

b → min factor n/a != a or 1

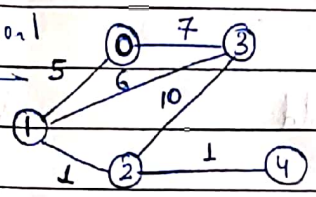
c → min factor $\frac{n}{ab}$!= a or b or 1

$-200(1+r)^2 + 100(1+r) + 100 = 0$

$200 + 100r = 200(1+r^2+2r)$

$200 + 100r = 200 + 200r^2 + 400r$

$r = 0$



$Z = (150\times4)\cdot(4\times6) = 150\times6$

$Z = (150\times6)\cdot(6\times1) = 150\times1$

Back →

o_e =

(0,0), (1,1), (1,3)

(4,0) (4,4)

dl → abs(x−y) == 0

dr → x+y = constant

ul → abs(x−y) == C

ur → x+y == C.

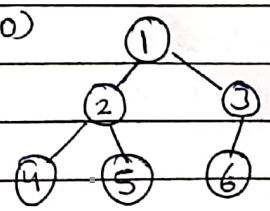
0 → (0,0)

dl =

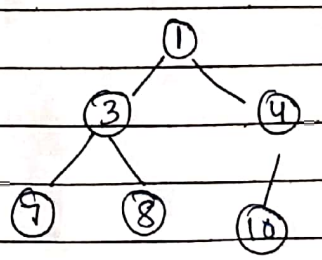
dr = 0 → (0,0)

ul =

ur =



⇒



	H_1	H_2	H_3	O/P
I/P	0			
0	0	0		0
0	0		0	0
0	0	0		
			0	
0	0	0		0
90×4	0			

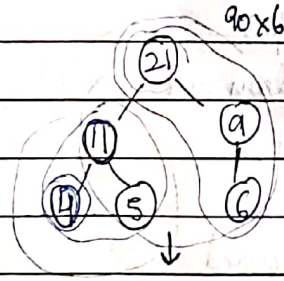
$W_0 = 4\times6$ $W_1 = 6\times3$ $W_2 = 3\times2$ $W_3 = 2\times3$

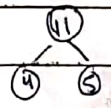
$W_1 =$ Z_0 90×6 Z_1 90×3 Z_2 90×2 Z_3 90,3

$er_2 = d_1$

$er_1 = y - o_1$

$d_1 = er_1 * sig(o_1)$





- Generators in Python
  any function that uses yield f^n are generators.

```
def genFunc():
    yield 1
    yield 2
    yield 3.

x = getFun()
print(x.next())
print(x.next()).
```

- Fibonacci using yield

```
a=0; b=1
while True:
    yield a
    a,b = b, a+b.
```

use of generators → handling streaming data.

- Python always passes value by reference.
- Maglev → Google's Network Load Balancer.
  ↳ a distributed software for doing load balancing → uses location and finds nearest servers.

(Q) The number of different balanced parenthesis of length $2n$.

- Fastest way to find id present in one file and not in the other
  cat f1 | f2 | Sort | --unique

- LCA
- LIS

Date
Page

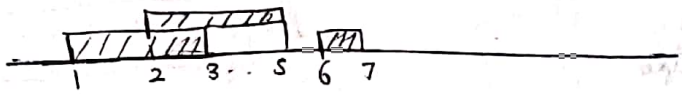
Java GS

(1) eCommerce
design currency converter.
idea: find way to segregate responsibilities

(Q) Design a Messaging System
(Q) what technical challenges have I faced till now and how did I managed to resolved them
↳ Imp.

(3)
(1) Serialization
↳ only 1 transaction at a time
no concurrency.
(2) Code of Factory Pattern

Scanned with CamScanner

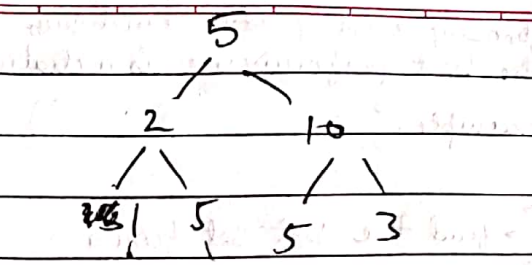


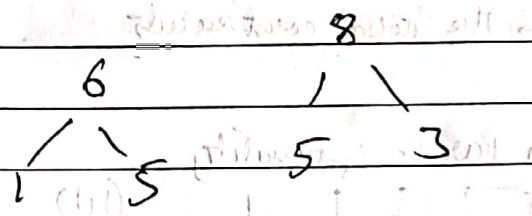
Date
Page

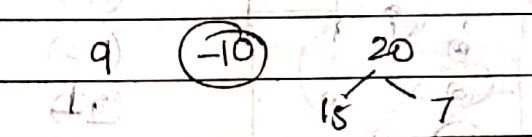
Story

The Happy Man's Shirt

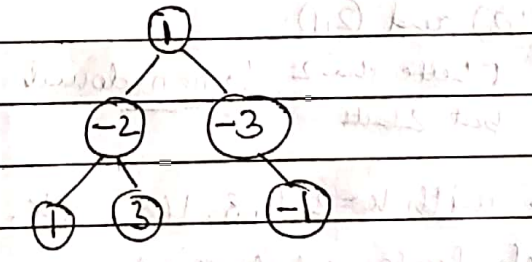
- are you in love, (arrange Marriage at any cost)
- Theaters, balls, concerts and singing were all useless.
- After studying the stars
- A priest tried to fool the King for his wealth. King refused.
- Met a neighbouring King → worried about the what would happen of his wealth once he dies.
- chasing the hair reached an open field. and met a young man.
- To test his → King offered him to be his friend → the young man refused.

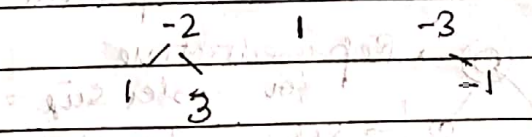




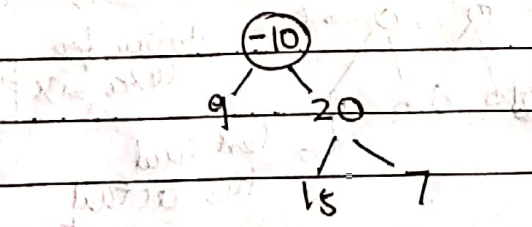


9 -10 15 20 7





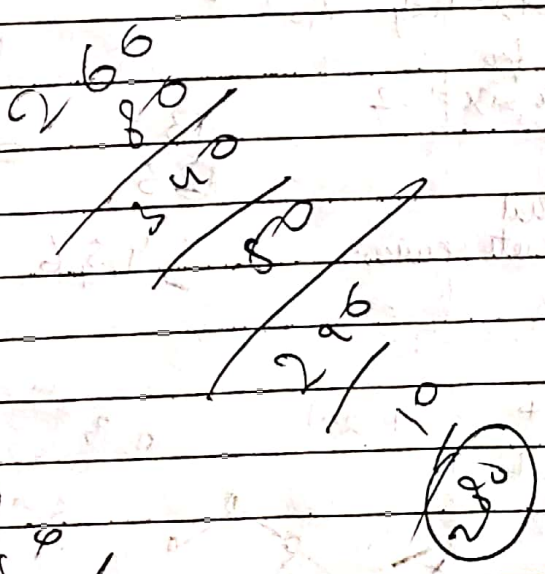
1 -2 3 1 -3 -1

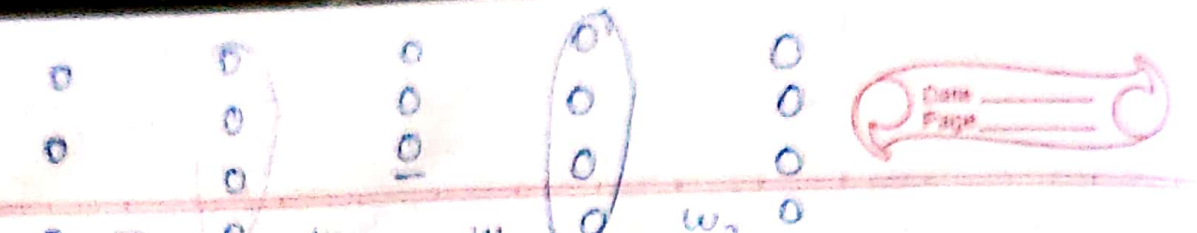


9 -10 15 20 7

Ab 260 Pr → 240

Ans 260





$e_0 =$ delta * w_1

delta * w_0 * dri(z_0(0))

on = delta * w_2

delta = on * dri(z_1(0))

err = delta * w_3

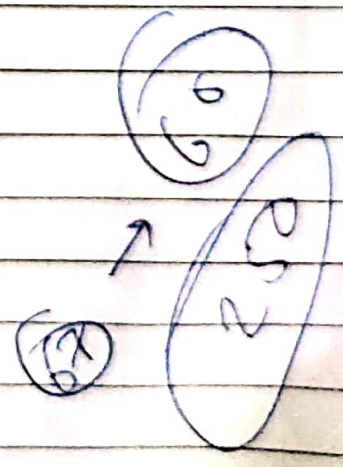
delta = err * dri(on * z_2(0))

notac

err = y - act O/P

delta = err * ~~opt~~ dsig(opt)

$$w_3 = w_3 - lr * \left[z_{i-2}[t] * \text{last calculated delta} \right]$$



268/4

270/40

Good After Noon Everyone

Make 2 x 3 x 3

Date
Page

Slide 1 → Nowadays we have a large no. of alternate services for a single task. So choosing a set of task which gives an user the best performance is a challenge.

Given Example.

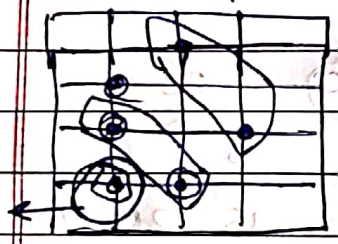
Slide 2 →

Challenge → find the best set from a very large pool of services.

Slide 3 → what are the selection criteria → The Global constraints

Slide 4 →

Explain Pareto optimality



(1,1)
(1,2)
(2,1)
(2,4)
(3,2)

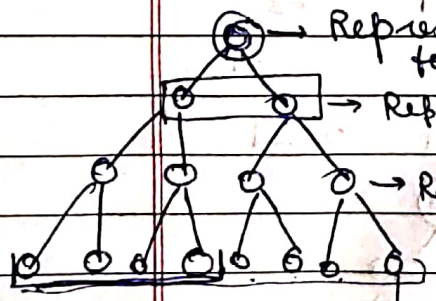
(1,1) > (2,2)

(1,2) and (2,1)

1 better than 2 but 2 better than 1 → non-domain

K-Means → with k = 2, 4, 8, 16, ... k.

for each cluster select a represent



→ Representative for cluster size = k=8

→ Represen for cluster size = k=4

→ Representative for cluster size = k=2

Last level the actual concrete services.

— — —

$\frac{5\times4}{2} + \frac{4\times3}{2} + \frac{3\times2}{2} + \frac{2\times1}{2}$

10 + 6 + 3 + 2

Here we decompose the global constraints into local one.

These local constraints are given to brokers → (serve as upper bound) and they return the best services.

Even, odd Query

3 2 7

pow(2, find(3,3))

↓

pow(7, find(4,3))

pow(3, (2,4))

↓

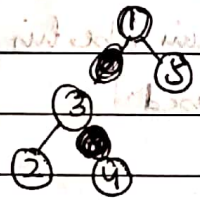
pow(2, (3,4))

↓

pow(5, (4,4))

↓

pow(7, 1)



(2,2,2)

(2,1)
(1,2)
(0,0)
(0,0)
(12,2)
(11,11)
(2,2)
(1,1)
(0,10)
(0,10)

9x9 = 81

1 3
2 3

2 3

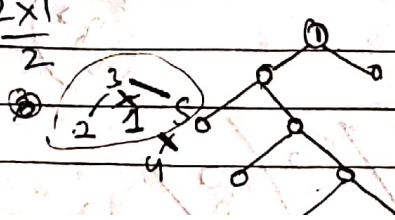
3, 4, 5, 6

4, 3
3, 2
6, 2
5, 3
6, 3
5, 2

$^2C_1\ ^3C_1 + ^2C_1\ ^3C_2 + ^2C_1\ ^3C_3$

2x3 + 2x3 + 2x1

3 → 2
4 → 3
5 → 3, 2
6 → 2, 3



Scanned with CamScanner

```
int partition (int a[], int low, int high)
{   int pivot = a[high].
    int i = (low - 1);
    for ( j = low ; j < high ; j++)
      { if (a[j] <= pivot)
        {   i++;
            swap (a[i], a[j]);
        }
    swap ( a[i+1], a[high])
    return i+1;
void quick sort (int a[], int low, int high)
{   if (low < high)
    {   int pi = partition(a, low, high)
        quicksort (a, low, pi-1)
        quick sort (a, pi+1, high)
    }
}.
```

1 2 3 4 5

1 2 0 1 2

$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$

$y = y_1 = mx - mx_1$

$mx - y + y_1 - mx_1 = 0$

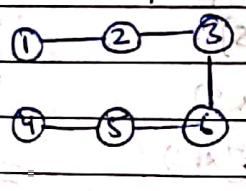
$\boxed{mx - y = mx_1 - y_1}$

for (i = x_1+1 ; i < x_2 ; i++)

$y = mi - mx_1 + y_1$

1 1 2 2 4 2 6 2 8

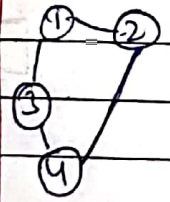
1 - 2 , 4,2



10

4 people → 10, 20, 30, 40

2 → {1,3}



$\frac{0-2}{4-0} = -\frac{1}{2}$

$y - 2 = -\frac{1}{2}x$

$2y - 4 + x = 0$

$\boxed{x + 2y = 4}$ → integer sol^n

$x > 0$ and $x < 4$

1 ✗, 2 ✓

if (s[i] == s[j])

dp[i][j] = dp[i+1][j-1] + 1

else

(8,16)

(1,9)

$y - 9 = \frac{16-9}{8-1}(x-1)$

$y - 9 = \frac{7}{7}(x-1)$

$y - 9 = x - 1$

$\boxed{x - y + 8 = 0}$

$\boxed{y - x = 8}$

10,2
11,3
12,4
13,5
14,6
15,7

$y - x = 8$

$x > 1$ and $x < 9$

Scanned with CamScanner

Stars

6

2 1 3

3 1

3 2

3 4

3 5

3 6

7

3 2 1

4 1

4 2

4 3

4 5

4 6

4 7

1 2 3 4 2 0 2

1 1 1

1 +

123 → 1,2,3 | 4,2,5 | 1,2,4 | 4,2,6 | 1,2,5 | 5,2,6 | 1,2,6 | 3,2,4 | 3,2,5 | 3,2,6

$\frac{5\times4}{2}=10$

abccba

aba

abba

Paths

5

2 1 3

1 2

2 3

3 4

4 5

6

1 2 3

1 2

2 3

3 4

4 5

5 6

[1 2] [5 6]

[4 4] [6 7 8]

[2 3 4 5]

[3×9] + [3×9]

(7×13) + (7×13)

7×7 + 13×13 + 4×4

169

49

16

234

234

468

Normal

6

3 2 1

1 2

1 3

2 7

3 4

3 5

3 6

7

1 2 3

1 2

1 3

2 4

2 5

3 6

3 7

1,3,13

3×4×14

12

14

128

14

168

1 2 3

1 2

169

49

16

234

+

• count Range sum: (dp version)

l = m.lower_bound(sum − upper)

r = m.lower_bound(sum − lower)

so count all b/w l to r in the map.

23

23

23

3 5 2 1 3 12

c=0

c=0

c=1

```
merge()
int i=l, j=m+1   vector<int> temp.
while (i<m and j<=r)
  {
  }
copy temp to a
for( i=l ; i<=r ; i++)
    a[i] = temp[i]

merge sort()
if (l<r)
  m = (l+r)/2
  merge sor(l,m)
  merge so(m+1,r)
  merge(l,m,r)
```

in inversion count

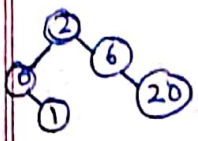
a[i] > a[j]

c += (m − i)

1 1 1

1 2 3

Scanned with CamScanner



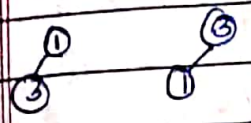
$4 = 0$

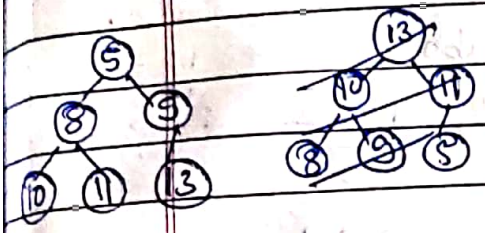
Date
Page

1 3 3 1 5, 8, 9, 10, 11, 13

(1,2) (3,1) 1

0 → 2





$\sqrt{x_3+7} + \sqrt{x_4+8}$

$()$

$\frac{1}{\sqrt{x_1+5}} = 0$

$\frac{}{x_2+6} = 0, \quad \frac{1}{2\sqrt{x_3+7}} = 0, \quad \frac{1}{2\sqrt{x_4+7}} = 0$

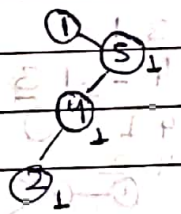
$\boxed{5, 8, 9, 10, 11, 13}$

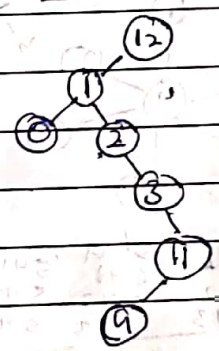
(0,10), (0,8), (0,9), (0,10) (0,11), (0,13)

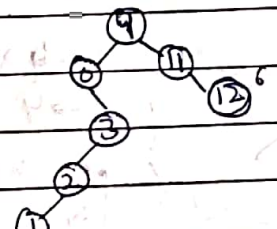
=A
VD
A

→ Document similarity
- Search engines
- Plagiarism Checkers

$\boxed{1\ 5\ 4\ 2\ 1}$

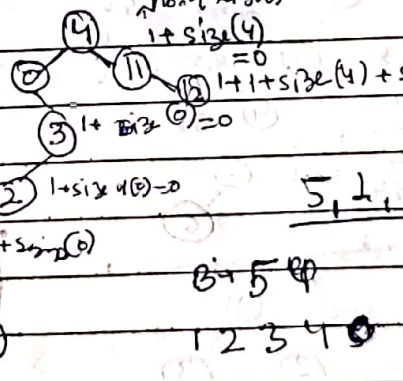




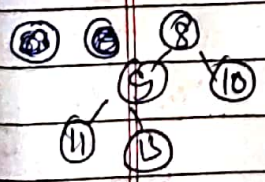


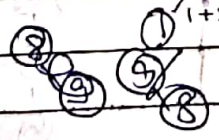
12, 1, 2, 3, 0, 11, 4

no. of nights



5, 1, 2, 4, 3





$2^1 + 2^2$

1 2 3 4

$6 + 2 \cdot 6 + 3 \cdot 6 \ldots n \times 6$

$\frac{n}{2}(2a + (n-1)d)$

$\frac{}{2} \rightarrow \frac{}{2}\ \frac{}{2}$

$2^1 + 2^2 + \ldots + 2^n$

$\frac{2}{2}[12+6] \Rightarrow 1$

a = 19

9 1 9 → 2

CDN's

$V_1 = (3, 2, 5)$

$V_2 = (1, 2)$

100

$V_1 . V_2$

① Star Graph

1 2 3 , 1 3 2

3,3 = 9

2 1 3

4,1,5 | 4,2,5 | 4,3,5
4,1,6 | 4,2,6 | 4,3,6
4,1,7 | 4,2,7 | 4,3,7

3,3

2 3 1

4,5,1 | 4,6,1 | 4,7,1
4,5,2 | 4,6,2 | 4,7,2
4,5,3 | 4,6,3 | 4,7,3

3 1 2

5,4,1 | 6,4,1 | 7,4,1
5,4,2 | 6,4,2 | 7,4,2
5,4,3 | 6,4,3 | 7,4,3

2 1 3
5 4 6
5 4 7
6 4 7

2 3 1
2 4 1 2 1 3
3 4 1
3 4 2

$S \to 4 \times 3 = 12$

$l \to 4$

$\begin{bmatrix} 3,2,4 \\ 3,2,5 \\ 3,2,6 \end{bmatrix}$, $\begin{bmatrix} 4,2,5 \\ 4,2,3 \\ 4,2,6 \end{bmatrix}$

5	2	7
5	7	0

0,0

2 1 3 on 7

5 4 7 6 4 7
5 4 6

Scanned with CamScanner

MD5
SHA

$$x_1 + 2x_2 + 3x_3 + 4x_4 = 0$$

Date
Page

$$f(x) = \sqrt{x_1+5} + \sqrt{x_2+6} + \sqrt{x_3+7} + \sqrt{x_4+8}$$

Maximise $f(x)$

intuition

$$\frac{df(x)}{dx} \rightarrow \frac{\partial f(x)}{\partial x_1} = \frac{1}{2\sqrt{x_1+5}} = 0$$

$a+b,\ b + a,\ a+b + a-b,\ b + a,b-a = x_1 \quad \frac{1}{2\sqrt{x_2+6}} = 0, \quad \frac{1}{2\sqrt{x_3+7}} = 0, \quad \frac{1}{2\sqrt{x_4+7}} = 0$

$4a, 4b = x, y$

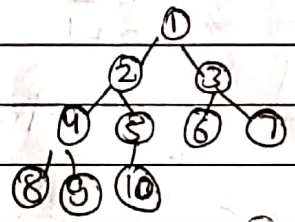
$a, b = \frac{x}{4}, \frac{y}{4}$

- Similar Items Mining
  - High dimensional Data
    - ↳ Data Reduction → PCA, SVD, ICA
  - clustering
  - Locality Sensitive Hashing (LSH)

→ Document similarity
- Search engines
- Plagiarism Checkers
- Recommenders

→ Route Similarity

- Set of documents – Vector representation.
  - ↳ angle between docs.
    - → Make the documents vectors



$D_1[\ ]$ = Word Freq.

$D_1[w_1] = \phi_1 \quad D_1[w_2] = \phi_2$

38, 3

(Q) why dot is not a good practice for document similarity?

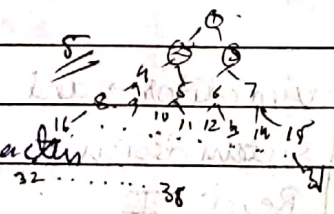
$$d(D_1, D_2) = \frac{D_1 D_2}{|D_1|\,|D_2|}$$

38

- Shingles (k-grams)

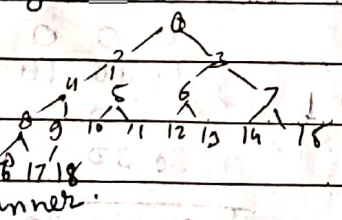
'the dog' k-consecutive characters

2-grams → th, he, e-, -d, do, og.



Shingles → Min Hashing → LSH

- Shingles: k-grams, frequency of k-grams
- Min Hashing: gen signatures (compressing)
- LSH: focussing on signatures in a local manner.



Scanned with CamScanner

Date
Page

20 + 21 + 14 = 55

12.5 + 21 + 14 = 12.5 + 35 = 47.5

32 + 42 + 30 = 124

(-5, 63) (-6, 58) (-7, 49) (-8, 38)

m = 30 26 21 15

h, 30 26 21 15

m 12.5 10 7 3.5

l = 12.5 10

1+2

m. 12.5 10 7 3.5 ≥ 0

2+1

3+3+1

5 7 9 5 27

5 7 0

a b

a(a) = 1 a(b+1) = -1

1 2 3 4 5 6

0 1 2 3 4 5

1 2 3 4 5 6

0 0 1 1 2 3

1 2 3 0 3

1 3 6 6 9

8 6 3 3

1 2 3 4 5

1 3 6 10 15

10 0 0 0 0

a(i) += a(i-1)

for i = 1 → n

a(i) += a(i-1)

10 10 10 10 10

10 10 10 0 0

- vim editor → and 10 codes within 2hrs.

System design.

React - II

13 → 10

25 → 20

12 → 30

0 0 0 0 0

10 10 10

20 20 20 20

30 30

40 60 30 20 20

0 0 0 0 0

10 -10

-20

30 -30

40 20 -30 -10 0

40 60 30 20 20

The Quick Brown Fox jumps over the lazy dog.

Scanned with CamScanner

n = 6 (6)

2 4 6 2 0 0

n = 5

1 2 3 1 2 i=0 0 1 2 3 4 5

5 3 6 9 1 0

1 2 3 0 1 2 3

n mid 2 3 2 1

L,R L,R 101

10 2-3

111

2 4 6 2 0

0 0 0 0 2

3 6 9 1 0 5 7 0 2-1 (1)

1 2 3 0 1

2 4 6 2 0 3 2 4 1 5 -1 = (2)

3 6 9 1 0

0 0 0 0 2

1 2 3 0 1

$\sqrt{x_1 + 5}$

1 2 3 4

5 6 7 8

5 (6) kg

-5

-5 -6 -7 -8

1.23 8.9 (12+21+32)

-5.67 2.34

b+1 < n

b > n

8.9 mid a(b+1) = -1

b_1 2.34 mid

5.67 -2.34

18 -6

1.23 8.90

5.67 -2.34

1 2 3 4

5 6 7 8

5+21+32 -5.67

12+5+32 17

(-5, 85) (-6, 58) (-7, 49) (-8, 38)

30×1 26×2 21×3 15×4 >=0

h= 30 26 21 15

12.5 10 7 -3.5 >=

Scanned with CamScanner

- Skyline Services

  Selection of which set of services which maximises the QoS is the challenge.

  Real and Synthetically composed datasets

- Global Optimization with Local Selection

  Find optimal decomposition of global QoS constraints into local constraints. Find best web services that suit the local constraints

  ↳ webservices changes dynamically.

a = 1 | 2 3 4 5 6 (2,2)

a = 1 → 1 2 1

1 2 1
0 1 2 → 0 1 2 = [0 1 2] − [1 2 1] * (0)
= (0 1 2)

~~2 7 8~~
0 3 6

[2 7 8] − [1 2 1] * 2
(2 7 8) − [2 4 2]
(0 3 6)

~~(0 0 0)~~ = [1 2 1] − [0 1 2] * 2

(1 2 1) − (0 2 4)
(1 0 −3)

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \\ -1 \end{bmatrix}$$

Praveen Kumar Sining

I've to make it to the top

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \\ -1 \end{bmatrix} [A^{-1}]$$

(1,1)

~~Date 16/09/19~~ (5 3) $\begin{bmatrix} x \\ y \end{bmatrix}$ = 7 (1,2)

$$\begin{bmatrix} x \\ y \end{bmatrix} = [7] \begin{bmatrix} 5 \\ 3 \end{bmatrix}^{-1}$$

rank = min (r, c) = 3

$$\begin{bmatrix} 2 & -1 & 3 \\ 1 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & 1 & 4 \end{bmatrix} \rightarrow$$

1 row → 1/2 = 0.5

1 2 3 4 5
1 2 3 4 5
0 0 0 0 0

1 −1 −1

1 1 1 1 1
1 2 2 1 3
0 0 0 0 0
1 −1 −1
1 1
1 1
1 −1 −1
2 3 3 4 3

2 −1 3
0 0 1
0 2 −1
0 1 4

2 −1 3
0 2 1
0 0 1
0 0 4

5 5 4 4 3