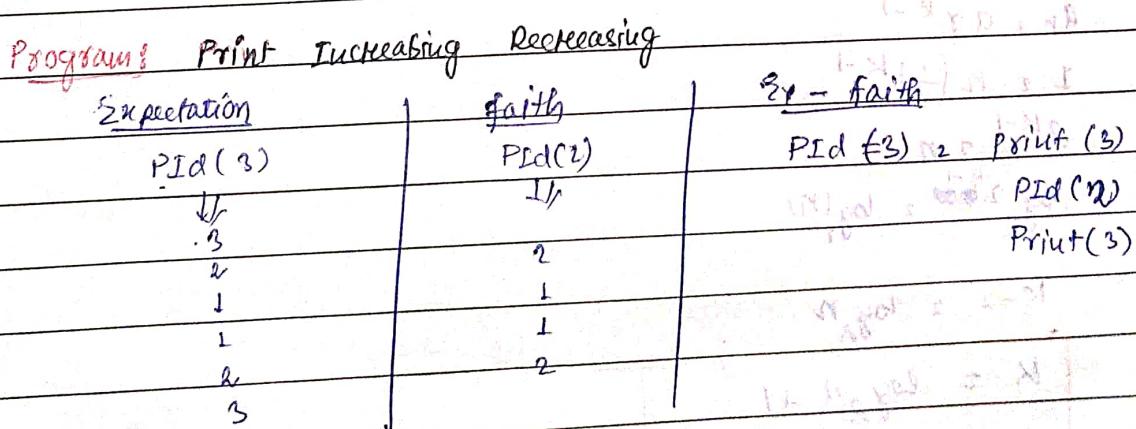
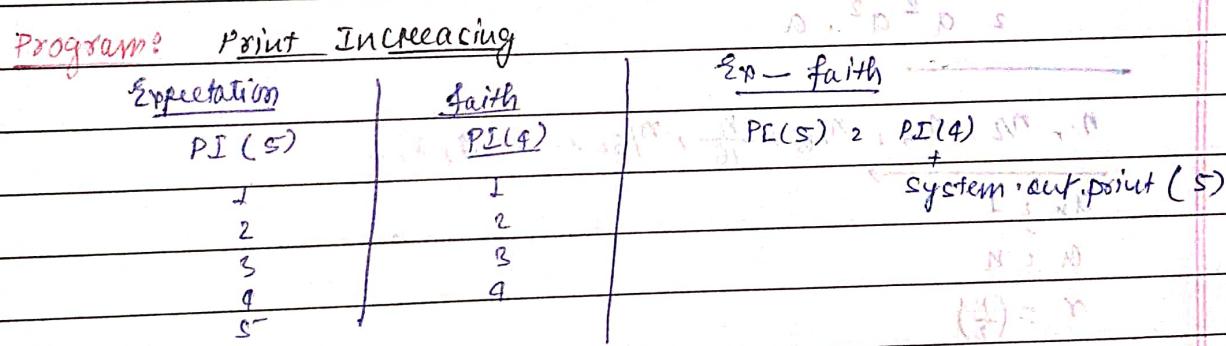
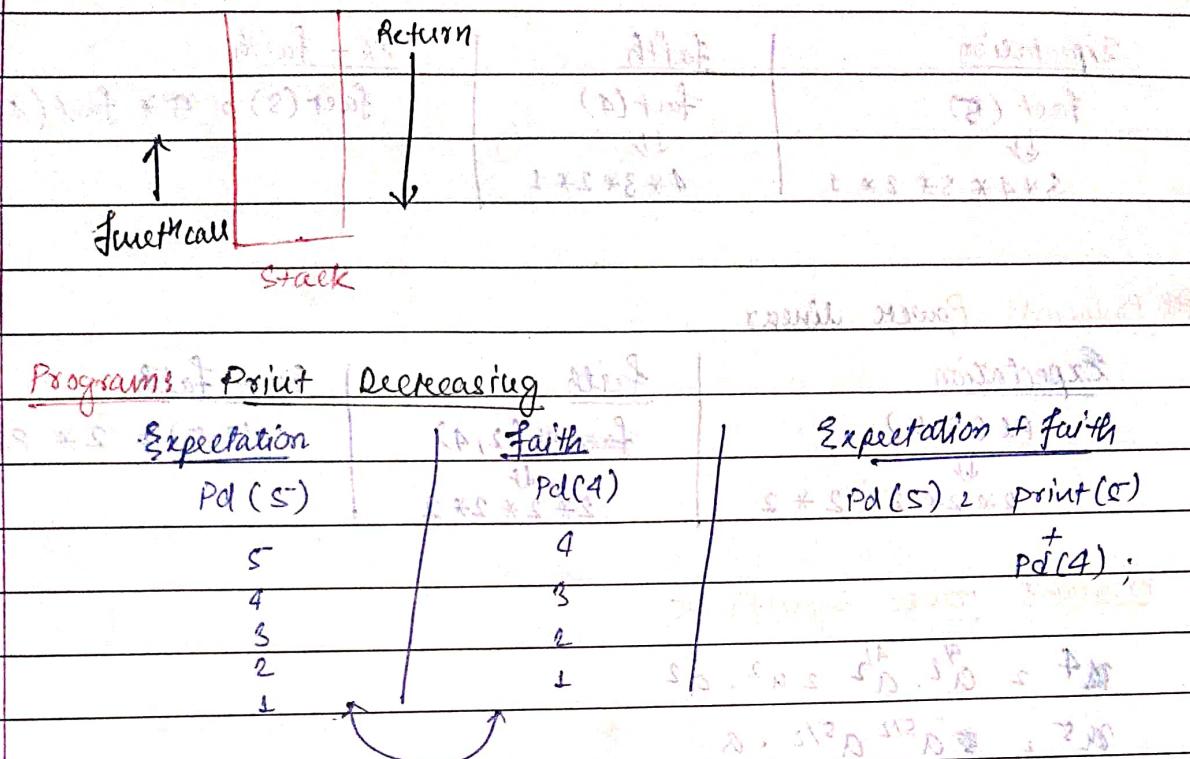


27/06/21



Problem factorial

Expectation

fact(5)



$$5 * 4 * 3 * 2 * 1$$

faith

fact(4)



$$4 * 3 * 2 * 1$$

Ex - faith

$$\text{fact}(5) = 25 * \text{fact}(4)$$

Problems Power linear

Expectation

PL(2,5)



$$2 * 2 * 2 * 2 * 2$$

faith

fact PL(2,4)



$$2 * 2 * 2 * 2$$

Ex - faith

$$\text{PL}(2,5) = 2 * \text{PL}(2,4)$$

Programs Power logarithmic

$$x^4 = a^2 \cdot a^2 \cdot a^2 \cdot a^2$$

$$x^5 = a^{5/2} \cdot a^{5/2} \cdot a$$

$$= a^2 \cdot a^2 \cdot a$$

$$n, n/2, n/4, n/8, \frac{n}{16}, n/32, \dots, 1, 2$$

$$a_k = 1$$

$$a = n$$

$$r = \left(\frac{1}{2}\right)$$

$$a_r, a_{r+1}$$

$$1 = n \left(\frac{1}{2}\right)^{K-1}$$

$$2^{K-1} = 2^n \left(\frac{1}{2}\right)^{K-1}$$

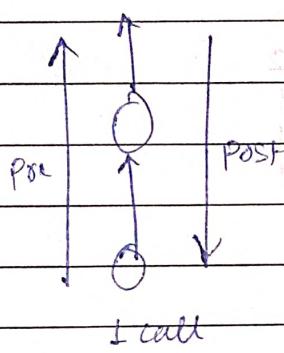
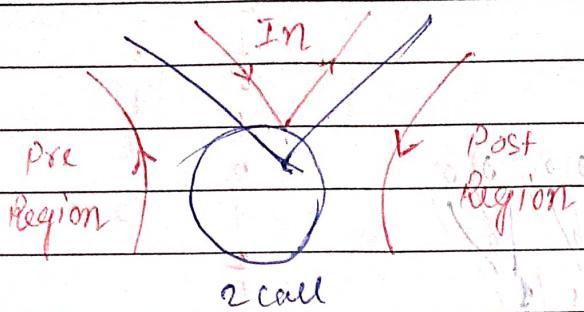
$$\log_2 2^{K-1} = \log_2 n$$

$$K-1 = \log_2 n$$

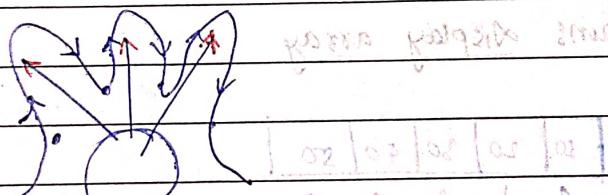
$$K = \log_2 n + 1$$

$$O(K) = O(\log_2 n)$$

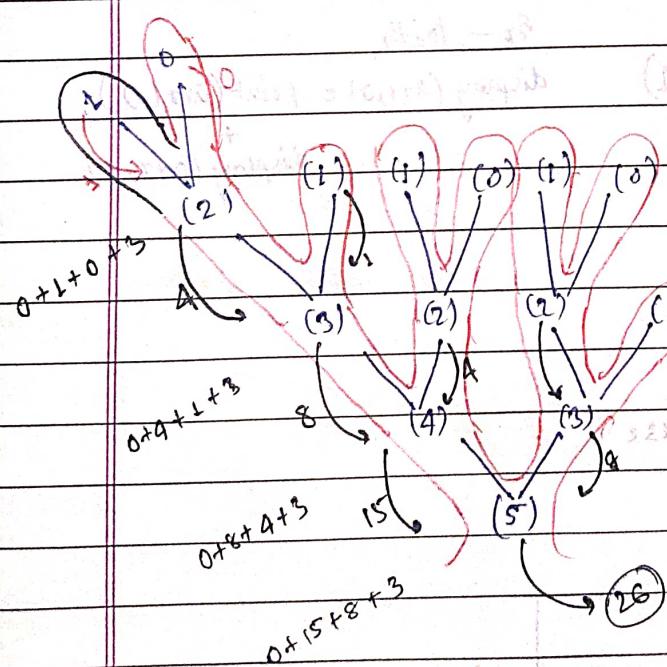
Euler function



1 call \rightarrow 2 visit
2 call \rightarrow 3 visit
 m call \rightarrow $(m+1)$ visit

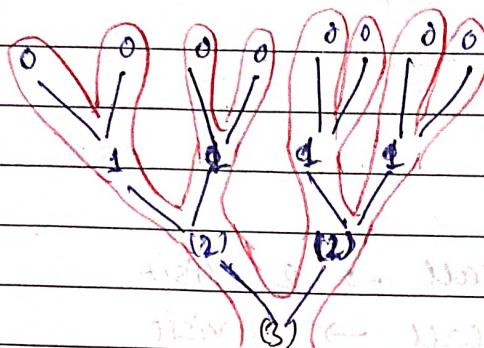


\rightarrow 4 visit



(0, 1) visit	(0, 2) Pre Stg 1	Post 1
(0, 3) visit	(0, 4) Pre Stg 2	Post 2
(0, 5) visit	(0, 6) Pre Stg 3	Post 3
(1, 2) visit	(1, 3) Pre Stg 4	Post 4
(1, 4) visit	(1, 5) Pre Stg 5	Post 5
(2, 3) visit	(2, 4) Pre Stg 6	Post 6
(2, 5) visit	(2, 6) Pre Stg 7	Post 7
(3, 4) visit	(3, 5) Pre Stg 8	Post 8
(3, 6) visit	(3, 7) Pre Stg 9	Post 9
(4, 5) visit	(4, 6) Pre Stg 10	Post 10
(4, 7) visit	(4, 8) Pre Stg 11	Post 11
(5, 6) visit	(5, 7) Pre Stg 12	Post 12
(5, 8) visit	(5, 9) Pre Stg 13	Post 13
(6, 7) visit	(6, 8) Pre Stg 14	Post 14
(6, 9) visit	(6, 10) Pre Stg 15	Post 15
(7, 8) visit	(7, 9) Pre Stg 16	Post 16
(7, 10) visit	(7, 11) Pre Stg 17	Post 17
(8, 9) visit	(8, 10) Pre Stg 18	Post 18
(8, 11) visit	(8, 12) Pre Stg 19	Post 19
(9, 10) visit	(9, 11) Pre Stg 20	Post 20
(9, 12) visit	(9, 13) Pre Stg 21	Post 21
(10, 11) visit	(10, 12) Pre Stg 22	Post 22
(10, 13) visit	(10, 14) Pre Stg 23	Post 23
(11, 12) visit	(11, 13) Pre Stg 24	Post 24
(11, 14) visit	(11, 15) Pre Stg 25	Post 25
(12, 13) visit	(12, 14) Pre Stg 26	Post 26
(12, 15) visit	(12, 16) Pre Stg 27	Post 27
(13, 14) visit	(13, 15) Pre Stg 28	Post 28
(13, 16) visit	(13, 17) Pre Stg 29	Post 29
(14, 15) visit	(14, 16) Pre Stg 30	Post 30
(14, 17) visit	(14, 18) Pre Stg 31	Post 31
(15, 16) visit	(15, 17) Pre Stg 32	Post 32
(15, 18) visit	(15, 19) Pre Stg 33	Post 33
(16, 17) visit	(16, 18) Pre Stg 34	Post 34
(16, 19) visit	(16, 20) Pre Stg 35	Post 35
(17, 18) visit	(17, 19) Pre Stg 36	Post 36
(17, 20) visit	(17, 21) Pre Stg 37	Post 37
(18, 19) visit	(18, 20) Pre Stg 38	Post 38
(18, 21) visit	(18, 22) Pre Stg 39	Post 39
(19, 20) visit	(19, 21) Pre Stg 40	Post 40
(19, 22) visit	(19, 23) Pre Stg 41	Post 41
(20, 21) visit	(20, 22) Pre Stg 42	Post 42
(20, 23) visit	(20, 24) Pre Stg 43	Post 43
(21, 22) visit	(21, 23) Pre Stg 44	Post 44
(21, 24) visit	(21, 25) Pre Stg 45	Post 45
(22, 23) visit	(22, 24) Pre Stg 46	Post 46
(22, 25) visit	(22, 26) Pre Stg 47	Post 47
(23, 24) visit	(23, 25) Pre Stg 48	Post 48
(23, 26) visit	(23, 27) Pre Stg 49	Post 49
(24, 25) visit	(24, 26) Pre Stg 50	Post 50
(24, 27) visit	(24, 28) Pre Stg 51	Post 51
(25, 26) visit	(25, 27) Pre Stg 52	Post 52
(25, 28) visit	(25, 29) Pre Stg 53	Post 53
(26, 27) visit	(26, 28) Pre Stg 54	Post 54
(26, 29) visit	(26, 30) Pre Stg 55	Post 55
(27, 28) visit	(27, 29) Pre Stg 56	Post 56
(27, 30) visit	(27, 31) Pre Stg 57	Post 57
(28, 29) visit	(28, 30) Pre Stg 58	Post 58
(28, 31) visit	(28, 32) Pre Stg 59	Post 59
(29, 30) visit	(29, 31) Pre Stg 60	Post 60
(29, 32) visit	(29, 33) Pre Stg 61	Post 61
(30, 31) visit	(30, 32) Pre Stg 62	Post 62
(30, 33) visit	(30, 34) Pre Stg 63	Post 63
(31, 32) visit	(31, 33) Pre Stg 64	Post 64
(31, 34) visit	(31, 35) Pre Stg 65	Post 65
(32, 33) visit	(32, 34) Pre Stg 66	Post 66
(32, 35) visit	(32, 36) Pre Stg 67	Post 67
(33, 34) visit	(33, 35) Pre Stg 68	Post 68
(33, 36) visit	(33, 37) Pre Stg 69	Post 69
(34, 35) visit	(34, 36) Pre Stg 70	Post 70
(34, 37) visit	(34, 38) Pre Stg 71	Post 71
(35, 36) visit	(35, 37) Pre Stg 72	Post 72
(35, 38) visit	(35, 39) Pre Stg 73	Post 73
(36, 37) visit	(36, 38) Pre Stg 74	Post 74
(36, 39) visit	(36, 40) Pre Stg 75	Post 75
(37, 38) visit	(37, 39) Pre Stg 76	Post 76
(37, 40) visit	(37, 41) Pre Stg 77	Post 77
(38, 39) visit	(38, 40) Pre Stg 78	Post 78
(38, 41) visit	(38, 42) Pre Stg 79	Post 79
(39, 40) visit	(39, 41) Pre Stg 80	Post 80
(39, 42) visit	(39, 43) Pre Stg 81	Post 81
(40, 41) visit	(40, 42) Pre Stg 82	Post 82
(40, 43) visit	(40, 44) Pre Stg 83	Post 83
(41, 42) visit	(41, 43) Pre Stg 84	Post 84
(41, 44) visit	(41, 45) Pre Stg 85	Post 85
(42, 43) visit	(42, 44) Pre Stg 86	Post 86
(42, 45) visit	(42, 46) Pre Stg 87	Post 87
(43, 44) visit	(43, 45) Pre Stg 88	Post 88
(43, 46) visit	(43, 47) Pre Stg 89	Post 89
(44, 45) visit	(44, 46) Pre Stg 90	Post 90
(44, 47) visit	(44, 48) Pre Stg 91	Post 91
(45, 46) visit	(45, 47) Pre Stg 92	Post 92
(45, 48) visit	(45, 49) Pre Stg 93	Post 93
(46, 47) visit	(46, 48) Pre Stg 94	Post 94
(46, 49) visit	(46, 50) Pre Stg 95	Post 95
(47, 48) visit	(47, 49) Pre Stg 96	Post 96
(47, 50) visit	(47, 51) Pre Stg 97	Post 97
(48, 49) visit	(48, 50) Pre Stg 98	Post 98
(48, 51) visit	(48, 52) Pre Stg 99	Post 99
(49, 50) visit	(49, 51) Pre Stg 100	Post 100
(49, 52) visit	(49, 53) Pre Stg 101	Post 101
(50, 51) visit	(50, 52) Pre Stg 102	Post 102
(50, 53) visit	(50, 54) Pre Stg 103	Post 103
(51, 52) visit	(51, 53) Pre Stg 104	Post 104
(51, 54) visit	(51, 55) Pre Stg 105	Post 105
(52, 53) visit	(52, 54) Pre Stg 106	Post 106
(52, 55) visit	(52, 56) Pre Stg 107	Post 107
(53, 54) visit	(53, 55) Pre Stg 108	Post 108
(53, 56) visit	(53, 57) Pre Stg 109	Post 109
(54, 55) visit	(54, 56) Pre Stg 110	Post 110
(54, 57) visit	(54, 58) Pre Stg 111	Post 111
(55, 56) visit	(55, 57) Pre Stg 112	Post 112
(55, 58) visit	(55, 59) Pre Stg 113	Post 113
(56, 57) visit	(56, 58) Pre Stg 114	Post 114
(56, 59) visit	(56, 60) Pre Stg 115	Post 115
(57, 58) visit	(57, 59) Pre Stg 116	Post 116
(57, 60) visit	(57, 61) Pre Stg 117	Post 117
(58, 59) visit	(58, 60) Pre Stg 118	Post 118
(58, 61) visit	(58, 62) Pre Stg 119	Post 119
(59, 60) visit	(59, 61) Pre Stg 120	Post 120
(59, 62) visit	(59, 63) Pre Stg 121	Post 121
(60, 61) visit	(60, 62) Pre Stg 122	Post 122
(60, 63) visit	(60, 64) Pre Stg 123	Post 123
(61, 62) visit	(61, 63) Pre Stg 124	Post 124
(61, 64) visit	(61, 65) Pre Stg 125	Post 125
(62, 63) visit	(62, 64) Pre Stg 126	Post 126
(62, 65) visit	(62, 66) Pre Stg 127	Post 127
(63, 64) visit	(63, 65) Pre Stg 128	Post 128
(63, 66) visit	(63, 67) Pre Stg 129	Post 129
(64, 65) visit	(64, 66) Pre Stg 130	Post 130
(64, 67) visit	(64, 68) Pre Stg 131	Post 131
(65, 66) visit	(65, 67) Pre Stg 132	Post 132
(65, 68) visit	(65, 69) Pre Stg 133	Post 133
(66, 67) visit	(66, 68) Pre Stg 134	Post 134
(66, 69) visit	(66, 70) Pre Stg 135	Post 135
(67, 68) visit	(67, 69) Pre Stg 136	Post 136
(67, 70) visit	(67, 71) Pre Stg 137	Post 137
(68, 69) visit	(68, 70) Pre Stg 138	Post 138
(68, 71) visit	(68, 72) Pre Stg 139	Post 139
(69, 70) visit	(69, 71) Pre Stg 140	Post 140
(69, 72) visit	(69, 73) Pre Stg 141	Post 141
(70, 71) visit	(70, 72) Pre Stg 142	Post 142
(70, 73) visit	(70, 74) Pre Stg 143	Post 143
(71, 72) visit	(71, 73) Pre Stg 144	Post 144
(71, 74) visit	(71, 75) Pre Stg 145	Post 145
(72, 73) visit	(72, 74) Pre Stg 146	Post 146
(72, 75) visit	(72, 76) Pre Stg 147	Post 147
(73, 74) visit	(73, 75) Pre Stg 148	Post 148
(73, 76) visit	(73, 77) Pre Stg 149	Post 149
(74, 75) visit	(74, 76) Pre Stg 150	Post 150
(74, 77) visit	(74, 78) Pre Stg 151	Post 151
(75, 76) visit	(75, 77) Pre Stg 152	Post 152
(75, 78) visit	(75, 79) Pre Stg 153	Post 153
(76, 77) visit	(76, 78) Pre Stg 154	Post 154
(76, 79) visit	(76, 80) Pre Stg 155	Post 155
(77, 78) visit	(77, 79) Pre Stg 156	Post 156
(77, 80) visit	(77, 81) Pre Stg 157	Post 157
(78, 79) visit	(78, 80) Pre Stg 158	Post 158
(78, 81) visit	(78, 82) Pre Stg 159	Post 159
(79, 80) visit	(79, 81) Pre Stg 160	Post 160
(79, 82) visit	(79, 83) Pre Stg 161	Post 161
(80, 81) visit	(80, 82) Pre Stg 162	Post 162
(80, 83) visit	(80, 84) Pre Stg 163	Post 163
(81, 82) visit	(81, 83) Pre Stg 164	Post 164
(81, 84) visit	(81, 85) Pre Stg 165	Post 165
(82, 83) visit	(82, 84) Pre Stg 166	Post 166
(82, 85) visit	(82, 86) Pre Stg 167	Post 167
(83, 84) visit	(83, 85) Pre Stg 168	Post 168
(83, 86) visit	(83, 87) Pre Stg 169	Post 169
(84, 85) visit	(84, 86) Pre Stg 170	Post 170
(84, 87) visit	(84, 88) Pre Stg 171	Post 171
(85, 86) visit	(85, 87) Pre Stg 172	Post 172
(85, 88) visit	(85, 89) Pre Stg 173	Post 173
(86, 87) visit	(86, 88) Pre Stg 174	Post 174
(86, 89) visit	(86, 90) Pre Stg 175	Post 175
(87, 88) visit	(87, 89) Pre Stg 176	Post 176
(87, 90) visit	(87, 91) Pre Stg 177	Post 177
(88, 89) visit	(88, 90) Pre Stg 178	Post 178
(88, 91) visit	(88, 92) Pre Stg 179	Post 179
(89, 90) visit	(89, 91) Pre Stg 180	Post 180
(89, 92) visit	(89, 93) Pre Stg 181	Post 181
(90, 91) visit	(90, 92) Pre Stg 182	Post 182
(90, 93) visit	(90, 94) Pre Stg 183	Post 183
(91, 92) visit	(91, 93) Pre Stg 184	Post 184
(91, 94) visit	(91, 95) Pre Stg 185	Post 185
(92, 93) visit	(92, 94) Pre Stg 186	Post 186
(92, 95) visit	(92, 96) Pre Stg 187	Post 187
(93, 94) visit	(93, 95) Pre Stg 188	Post 188
(93, 96) visit	(93, 97) Pre Stg 189	Post 189
(94, 95) visit	(94, 96) Pre Stg 190	Post 190
(94, 97) visit	(94, 98) Pre Stg 191	Post 191
(95, 96) visit	(95, 97) Pre Stg 192	Post 192
(95, 98) visit	(95, 99) Pre Stg 193	Post 193
(96, 97) visit	(96, 98) Pre Stg 194	Post 194
(96, 99) visit	(96, 100) Pre Stg 195	Post 195
(97, 98) visit	(97, 99) Pre Stg 196	Post 196
(97, 100) visit	(97, 101) Pre Stg 197	Post 197
(98, 99) visit	(98, 100) Pre Stg 198	Post 198
(98, 101) visit	(98, 102) Pre Stg 199	Post 199
(99, 100) visit	(99, 101) Pre Stg 200	Post 200
(99, 102) visit	(99, 103) Pre Stg 201	Post 201
(100, 101) visit	(100, 102) Pre Stg 202	Post 202
(100, 103) visit	(100, 104) Pre Stg 203	Post 203
(101, 102) visit	(101, 103) Pre Stg 204	Post 204
(101, 104) visit	(101, 105) Pre Stg 205	Post 205
(102, 103) visit	(102, 104) Pre Stg 206	Post 206
(102, 105) visit	(102, 106) Pre Stg 207	Post 207
(103, 104) visit	(103, 105) Pre Stg 208	Post 208
(103, 106) visit	(103, 107) Pre Stg 209	Post 209
(104, 105) visit	(104, 106) Pre Stg 210	Post 210
(104, 107) visit	(104, 108) Pre Stg 211	Post 211
(105, 106) visit	(105, 107) Pre Stg 212	Post 212
(105, 108) visit	(105, 109) Pre Stg 213	Post 213
(106, 107) visit	(106, 108) Pre Stg 214	Post 214
(106, 109) visit	(106, 110) Pre Stg 215	Post 215
(107, 108) visit	(107, 109) Pre Stg 216	Post 216
(107, 110) visit	(107, 111) Pre Stg 217	Post 217
(108, 109) visit	(108, 110) Pre Stg 218	Post 218
(108, 111) visit	(108, 112) Pre Stg 219	Post 219
(109, 110) visit	(109, 111) Pre Stg 220	Post 220
(109, 112) visit	(109, 113) Pre Stg 221	Post 221
(110, 111) visit	(110, 112) Pre Stg 222	Post 222
(110, 113) visit	(110, 114) Pre Stg 223	Post 223
(111, 112) visit	(111, 113) Pre Stg 224	Post 224
(111, 114) visit	(111, 115) Pre Stg 225	Post 225
(112, 113) visit		

Problems: zig Zag.



3	2	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	3	2	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	3	2	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1

function call & return
Recursion in Array

Program: display array

10	20	30	40	50
0	1	2	3	4

→ 10 → 20 → 30 → 40 → 50

Expectation

display (arr, 0)

↓

10

20

30

40

50

Faith

display (arr, 1)

↓

20

30

40

50

Exp - faith

display (arr, 0) = print (arr(0))

+
display (arr, 1)

Program:

display Array in Reverse

10	20	30	40	50
0	1	2	3	4

Exp

display (arr, 0)

↓

50

40

30

20

10

Faith

display (arr, 1)

↓

50

40

30

20

10

display (arr, 0) = display (arr, 1)

+

print (arr(0))



Date 28/06/2021

Date 1/1	
Page	RANKA

Program 1 Max of Array

20	10	20	4	100
0	1	2	3	4

find max element

Defining max (D-4) = max0

max = 0;

if (max < arrs[0]) {

 max = arrs[0];

}

2	3	1
200	200	200

Problem 2 find element in Array

10	25	5	1	9	7	2	6	15
0	1	2	3	4	5	6	7	8

Expectation

FE(arr, 0, 7)

↓

return idx 5

faith(arr, 1, 7)

↓

check element

from index 1 to length-1

with d=7

↓

if find return idx

else return -1

Ex-faith

FE(arr, 0, 7) =

 if (arr[0] == 7)

 return idx;

else

 FE(arr, 1, 7)

Problem 3 first index

10	25	5	1	9	7	2	6	15
0	1	2	3	4	5	6	7	8

Problem 4 least index

34	19	3	2	19	6	19	7	8
0	1	2	3	4	5	6	7	8

d=19

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

34	19	3	2	19	6	19	7	8
0	1	2	3	4	5	6	7	8

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

6

↓

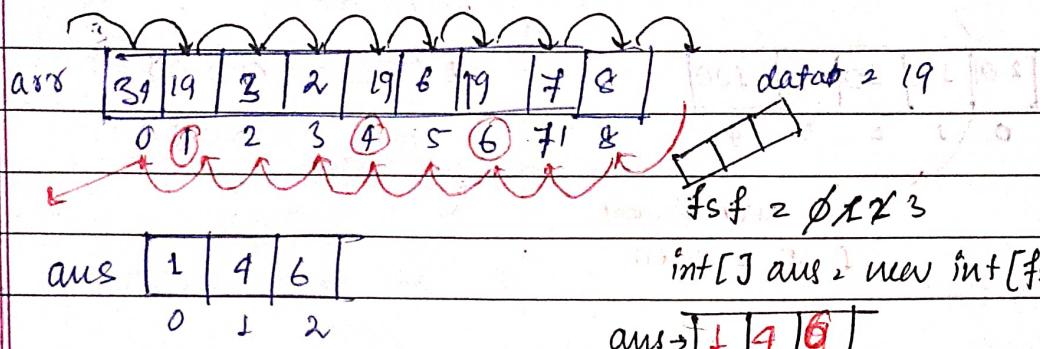
6

↓

6

↓

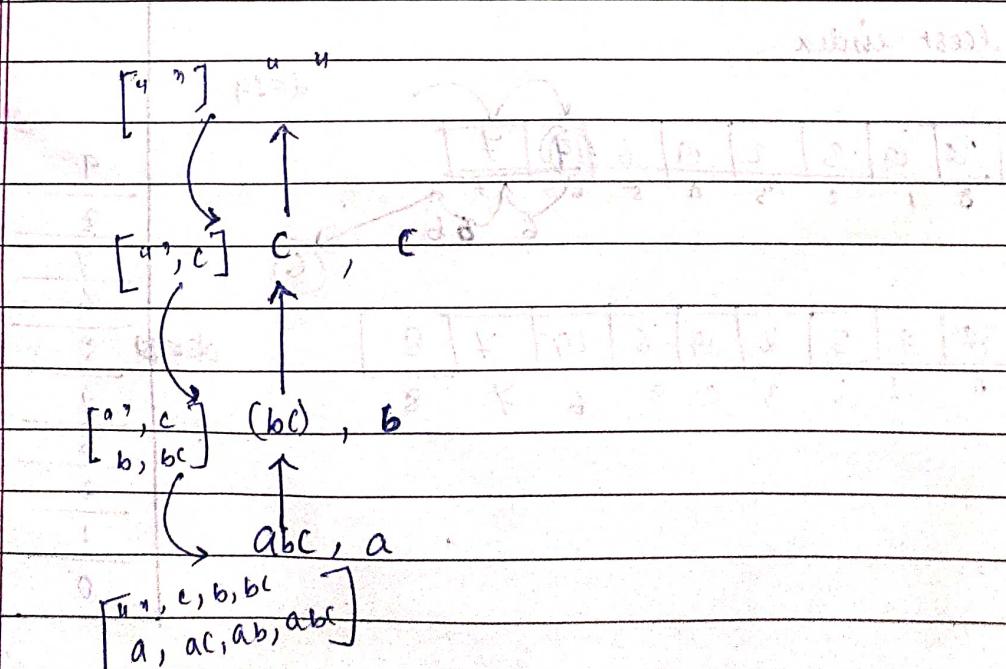
6

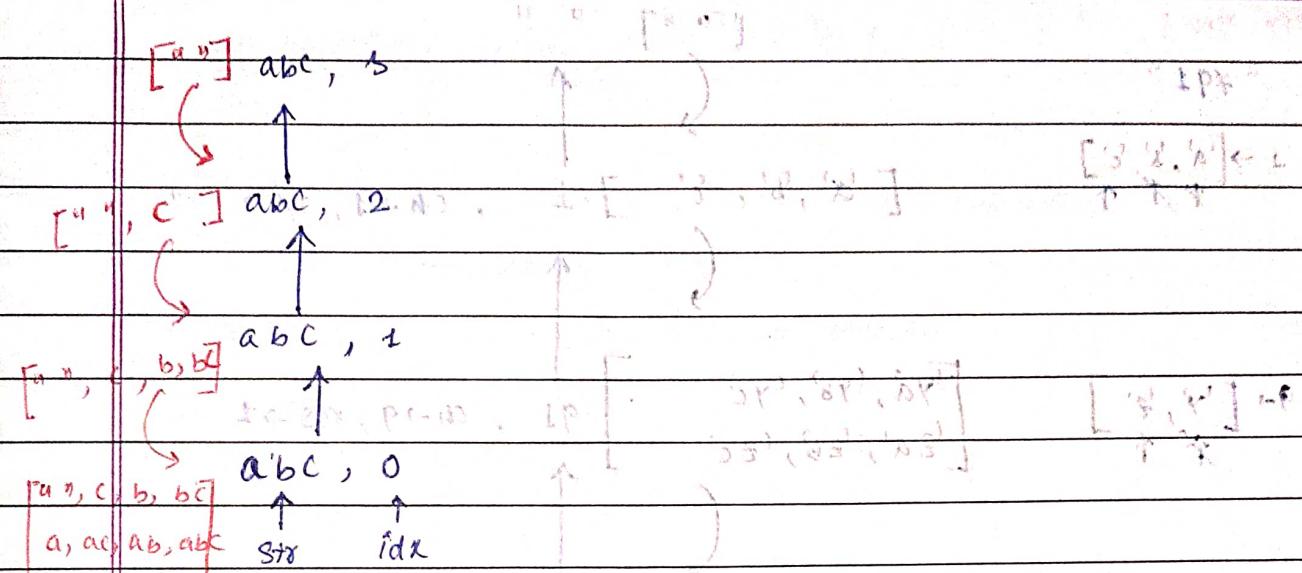
Problems All indices of ArrayRecursion in ArrayListProblems Get subsequence

abc

["", c, bc, b, a, ac, ab, abc]

Expectation	Faith	Expectation + faith
getss("abc")	getss("bc")	getss("ab") = getss("bc")
["", c, bc, b, a, ac, ab, abc]	["", c] [b, bc]	[] + getss("bc")



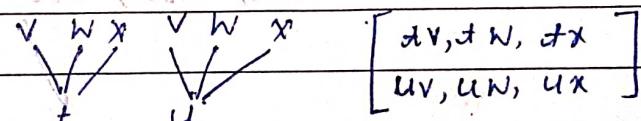


✓ Problems: Get Key PAD combination

input \Rightarrow 7 8

$\gamma \rightarrow t\bar{u}$

$g \rightarrow VWX$

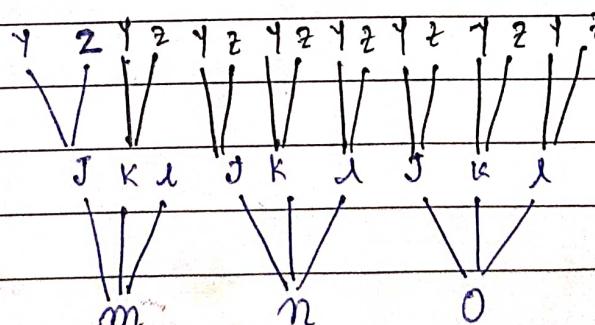


549

5 → MnO

4 → JKΛ

g → γZ



m_{JY}, m_{JZ}, m_{KY}, m_{KZ}, m_{LJ},
m_{LZ}, n_{JY}, n_{JZ}, n_{KY}, n_{KZ},
n_{LJ}, n_{LZ}, o_{JY}, o_{JZ}, o_{KY},
o_{KZ}, o_{LJ}, o_{LZ}

m_J, m_K, m_L, m_J, n_K, m_L
O_J, O_K, O_L

2008年7月22日



Code with time complexity

28/06

Power logarithmic $T(n)$ int powerBtr (int a, int b) {

if ($b == 0$) {

① return 1;

else int smallAns = powerBtr(a, $b/2$); $\rightarrow T(n/2)$ unit see

smallAns * = smallAns; \rightarrow ① unit seen = \Rightarrow 2nd repeat

return $b \% 2 == 0$? smallAns : smallAns * a; \rightarrow 1 + 1 + 2 unit ~~base~~ second

Time complexity (Mathematical)

$$T(n) = 1 + T(n/2) + 1 + 2$$

$$T(n) \approx 2 + T(n/2)$$

$$T(n/2) \approx 2 + T(n/4)$$

$$T(n/4) \approx 2 + T(n/8)$$

$$T(n/8) \approx 2 + T(n/16)$$

$$\vdots \quad ; \quad \vdots \\ T(1) \approx 2 + T(0)$$

$$T(n) \approx \sum_k^{\infty} 2 + T(0)$$

$$T(n) \approx \sum_k^{\infty} 2 + 1$$

$$\approx 4(\log_2 n + 1) + 1$$

$$\approx 4 \log_2 n + 4 + 1$$

$$\boxed{T(n) \approx 4 \log_2 n + 5}$$

$$O(T(n)) \approx O(4 \log_2 n + 5)$$

$$\boxed{O(T(n)) \approx O(\log_2 n)}$$

AP

$$n^{\text{th}} \text{ term} \approx a + (n-1)d$$

$$S_n = \frac{n}{2} [2a + (n-1)d]$$

$a \rightarrow$ first term

$d \rightarrow$ common diffⁿ

GP

$$a_n = a \cdot r^{n-1}$$

$$S_n = \frac{a(r^n - 1)}{r - 1} \quad [r < 1] \quad r \rightarrow \text{common ratio.}$$

$$\frac{1 - r^n}{1 - r} + 1 \quad [r < 1]$$

$$\frac{a(r^n - 1)}{r - 1} + [r > 1]$$

$$\frac{a}{1 - r} \quad [r < 1]$$

$T(n)$

(1) int powerBtr2(int a, int b) {

① if ($b == 0$)

return 1;

$T(n/2)$

$T(n/2) * T(n/2)$ int smallAns = powerBtr2(a, b/2) * powerBtr2(a, b/2);

② return $b \% 2 == 0$? smallAns : smallAns * a;

}

$$\rightarrow T(n) = 3 + T(n/2) * T(n/2)$$

$$= 3 + 2T(n/2)$$

$$= 2 \cdot 3 + 4T(n/4)$$

$$= 2^2 \cdot 3 + 8T(n/8)$$

$$= 2^3 \cdot 3 + 16T(n/16)$$

$$\vdots$$

$$T(n) = (1 + 2 + 2^2 + 2^3 + \dots) \cdot 3$$

$$= 2^{k-1}$$

$$= 2^{\log_2 n} - 1$$

$$T(n) = O(n) - 1$$

Optimization

$$O(T(n)) = O(n)$$

(2) int fibo(int n) {

① if ($n <= 1$) :

return n;

return fibo(n-1) + fibo(n-2)

$$= T(n-1) + T(n-2)$$

Mathematical

$$T(n) = 1 + T(n-1) + T(n-2)$$

~~Worst Case~~ $\Rightarrow T(n) < 1 + 2T(n-1)$

$$2T(n) < 2 + 4T(n-2)$$

$$4T(n-2) < 4 +$$

$$S_k = \frac{a(2^k - 1)}{2^k - 1}$$

$$= 1$$

$$= 2^{k-1}$$

$$= 2^{\log_2 n} - 1$$

Time complexity (Mathematical)

$$T(n) = 1 + T(n-1) + T(n-2)$$

for worst case

$$T(n) \leq 1 + T(n-1) + T(n-1)$$

$$T(n) \leq 1 + 2T(n-1)$$

$$2T(n-1) \leq 2 \cdot 1 + 2^2 T(n-2)$$

$$2^2 T(n-2) \leq 2^2 \cdot 1 + 2^3 T(n-3)$$

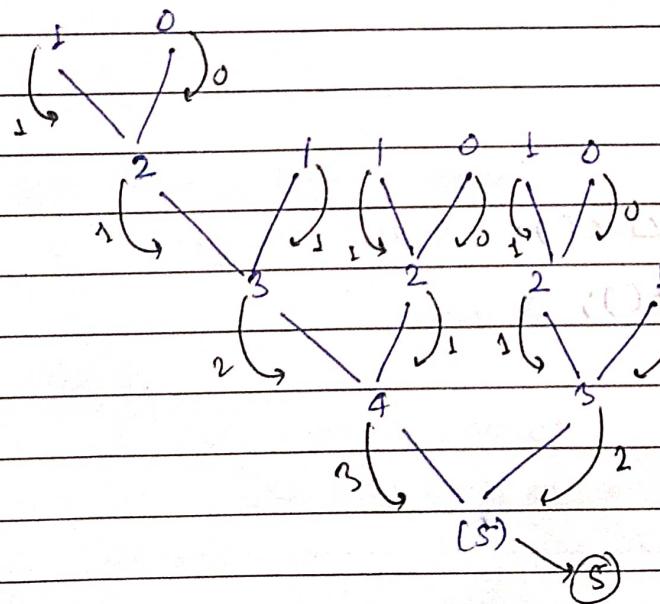
$$\therefore \alpha_{K-1}$$

$$T(n) \leq (1 + 2 + 2^2 + 2^3 + \dots)^{(k)} + T(0) \quad \Rightarrow (2)^{k-1} (1 + 2^{k-1})$$

$$\leq (2^k - 1) + 1$$

$$+ \leq 2^n - 1 + 1$$

$$\boxed{T(n) \leq 2^n}$$



Problems: Tribonacci Number.

$$T(n) = T(n-1) + T(n-2) + T(n-3)$$

```
int tribonacci (int n) {
    if (n <= 2)
        return n = 2 ? 1 : n;
    return tribonacci(n-1) + tribonacci(n-2) + tribonacci(n-3);
}
```

Mathematical

$$T(n) \leq 1 + 3T(n-1)$$

$$(1+3)^n \geq 3^n$$

String

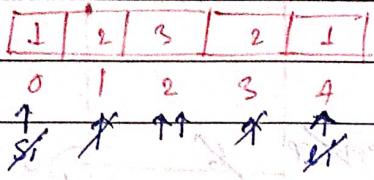
members

abc def ghi

String str1 = scn.nextLine(); // abc def ghi

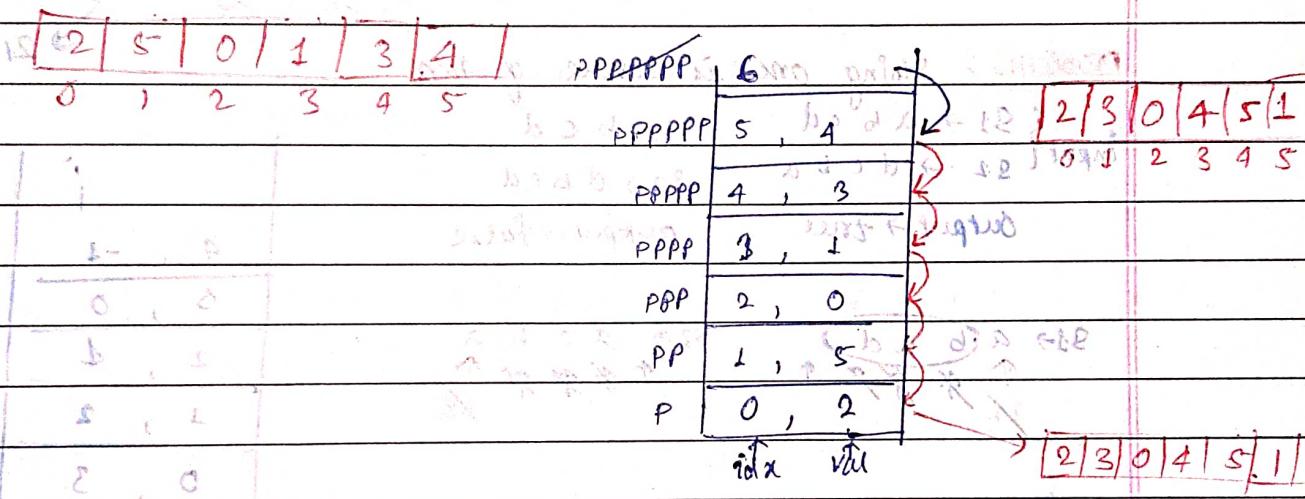
String str2 = scn.nextLine(); // abc

Problems 8 Palindrome using recursion (Array)



DPF	2, 2	→ true
DP	1, 3	→ true
P	0, 4	→ true

Problems: Inverse of an Array using Recursion.



Problems: Sum of Digit in String

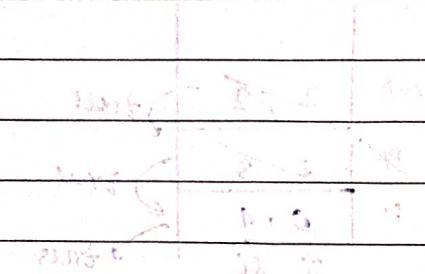
str = "23456789"
0 1 2 3 4 5 6 7 → sum = 42

faith &
 ① index 1 → str.length() - 1
 "3456789" → sum = 42
 ② convert '3' to number

Expectation	Faith	Exp + Faith
① sum(str, 0)	sum(str, 1)	sum(str, 0) = sum(str, 1)
↓ 44	↓ 42	+

day num's

1 2 3 4 5 6
0 1 2 5 4 5



PPPPPPP	6	0
PPPPP	6 + 0 = 6	
PPPPP	5 + 6 = 11	
PPPP	4 + 11 = 15	
PPP	18	
PP	20	
P	21	

Problems 8 String one is reverse of two

Input: $S_1 \rightarrow a b c d$ $S_1 \rightarrow a b c d$
 $S_2 \rightarrow d c b a$ $S_2 \rightarrow d b c a$
 Output \rightarrow true Output \rightarrow false

$S_1 \rightarrow a b c d$ $S_2 \rightarrow d c b a$
 ↓↑ ↑↓
 0 1 2 3 0 1 2 3

4, -4	→ true
5, 0	→ true
2, 4	→ true
1, 2	→ true
0, 3	→ true

faithless because sockets

Problem: Separate Duplicate say '*'

PEPPEPCODING

Output: PEP*PEPCODING

faith

EPPEPCODING

(EP*)PEPCODING

① If P & E are different character

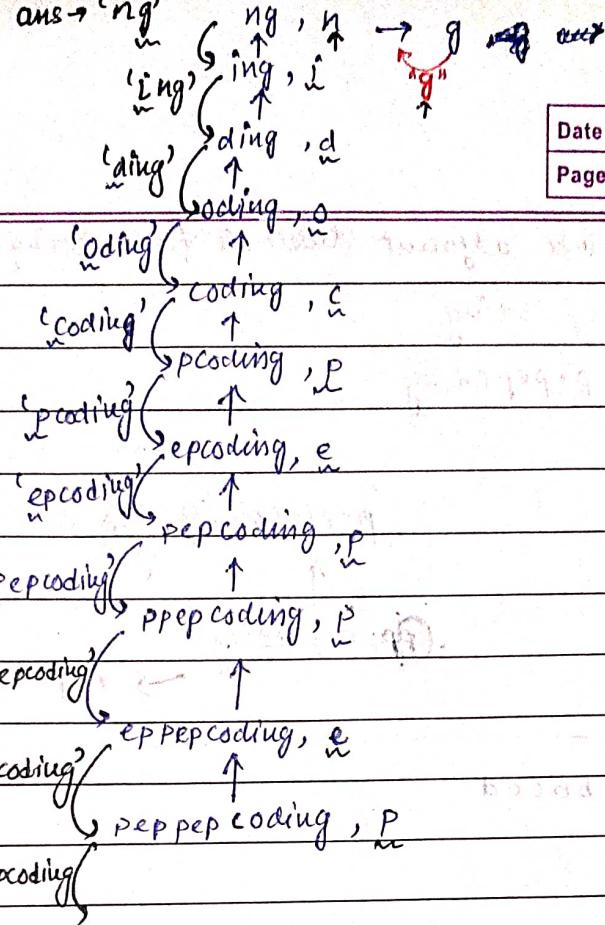
PEPPEPCODING, (P)

PEP*PEPCODING

② If suppose P = E, then

E * E * P * PEPCODING

str2 "PEPPING"



method 2

(way up)

PepperCoding

(8, e*ep*pepcod)

(7, e*ep*pepc)

(6, e*ep*pep)

(5, e*ep*pe)

(4, e*ep*p)

(3, e*ep*)

(2, e*)

(1, e*)

(0, "

idx ans

③ 'e*ep*pepcoding' + 'g' → (print) return

(11, e*ep*pepcodin)

(10, e*ep*pepcodi)

(9, e*ep*pepcod)

(8, e*ep*pepcod)

return false

true case

Store Karange



problem: Remove adjacent duplicate from string.

str → PEPPEP coding

output : pepcoding

P

pppepcoding → pepcoding
↑↑
① (pppepcoding) → Pepcoding

str → aaabbcccd

d (d)
↑
cd (cd), c
↑
cd (ccd), c
↑
cd (bccd), b
↑
bcd (bcd),
↑
bccd (bccd), b
↑
bcd (bcd),
↑
abced (abced), a
↑
abcd (abcd),
↑
aaabbcccd, a
↑
abcd (abcd),
↑



Date: 30/06

Date: / /

Page



Program: Given character Get Subsequence (By return type & array up)

a b c

["] , c, b, bc, a, ac, ab, abc]

b c

a b c , a

dry run

" ") [a "]

c , c [" c a " , c]

b c , b [" b " ; b c]

a b c , a [" a " , c , b , b c]

(1095)

(1-RST + 0-NRT + 1 - RST)

(1-NRT + Dec(0)) +

(1-RST + Dec(1-RST))

(1-RST + Dec(1-RST)) +

(1-RST + Dec(1-RST)) = (1095)

④ way up

P S V printSS (String str, int idx, String ans) {

if (idx == size) { print (ans); return; }

printSS (str, idx+1, ans + str.charAt(idx));

printSS (str, idx+1, ans);

1 + 1 - 8 =

abc ab ac' a bc' b c
 ab a b a c

a b c
 0 1 2

(2) + (2)(2)(2-1) + 10 (10)

Remember

* Formula for finding time complexity.

public static void printss (String str, int idx, String ans) {

 if (idx == str.length()) {

 System.out.println(ans);

 return;

}

 Printss(str, idx+1, ans + str.charAt(idx));

 Printss(str, idx+1, ans);

}

$$T(n) = 1 + T(n-1) + T(n-1)$$

$$T(n) = 2^0 + 2T(n-1)$$

$$2T(n-1) = 2^1 + 2^2 T(n-2)$$

$$2^2 T(n-2) = 2^2 + 2^3 T(n-3)$$

⋮

⋮

⋮

$$T(n) = (2^0 + 2^1 + 2^2 + \dots + 2^n) + T_0$$

$$= 2^{n+1} - 1 + 1$$

$$= 2^{n+1}$$

$$\frac{2^n - 1}{2 - 1} = 2^n - 1$$

Note:

* finding for time complexity.

$$T(n) = (\text{Work done in pre } + \text{post area}) \times \text{Height} + (\text{no of calls})^{\text{Height}}$$

work done in pre area = 0
in in post n = 20

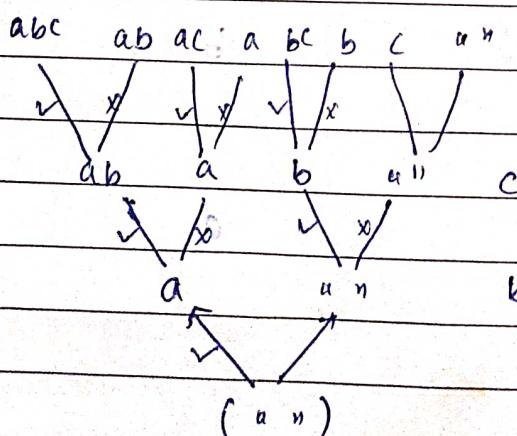
Height = n (for n element)

no of calls = 2

$$\therefore T(n) = (0+0) \times n + (2)^n$$

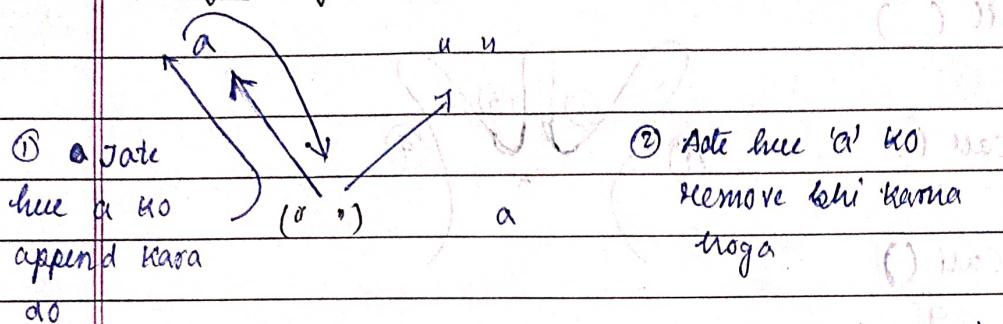
$$\boxed{T(n) = 2^n}$$

height



$$\boxed{T(n) = 2^n}$$

using string builder



p s v pointers_02(String str, int idx, String ans) {

if (idx == str.length()) {

System.out.println(ans);

} return;

ans.append(str.charAt(idx));

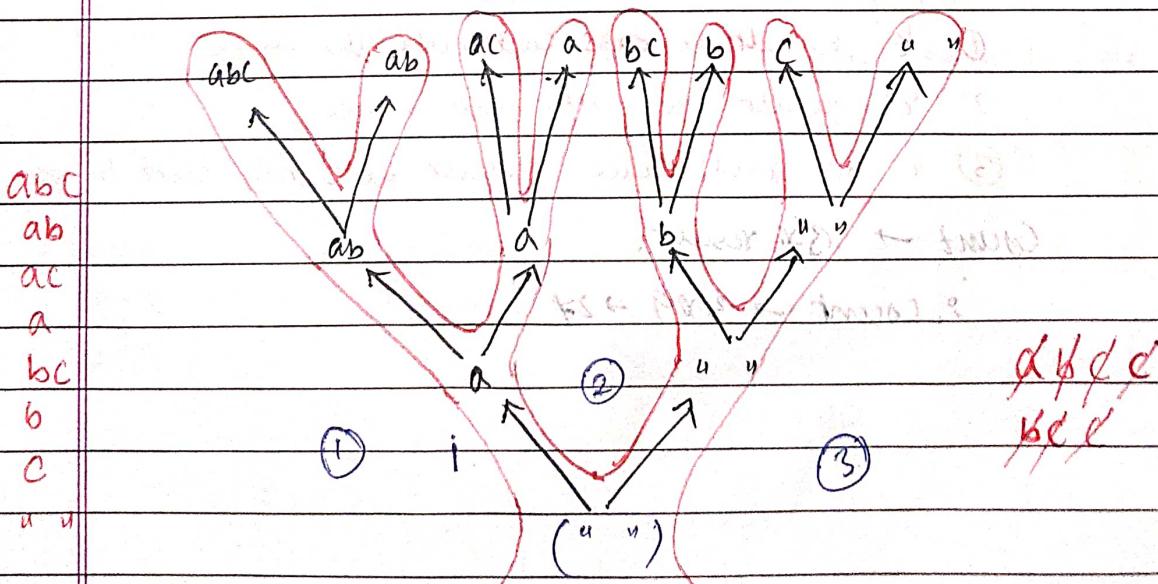
printSS_02(str, idx+1, ans);

ans.deleteCharAt(idx);

printSS_02(str, idx+1, ans);

}

③



Region

①
call ()

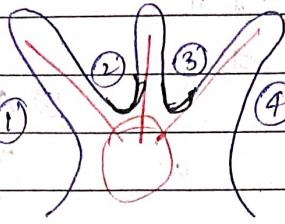
②

call ()

③

call ()

④



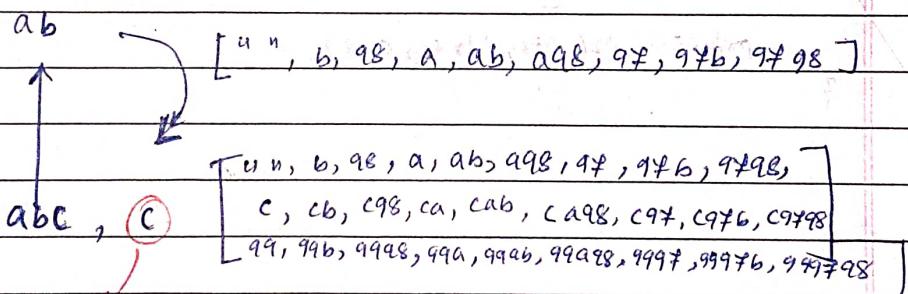
Date
10/7/2021

problem: Ascii subsequence

→ Str → [ab]

output → [" ", b, 98, a, ab, a98, 97, 97b, 9798]

faith



c ka pass tin option honge

① 'c' result ke pass sare add nahi honge

② 'c' result ke sare add honge

③ 'c' ka ascii value result ke sare add honge:

Count → 3 × 9 → 27

i. Count → 3 × 9 → 27

Problem: Get key pad combination

$0 \rightarrow 1$ $5\#6 \rightarrow$

$1 \rightarrow abc$

$2 \rightarrow def$

$3 \rightarrow ghi$

$4 \rightarrow jkl$

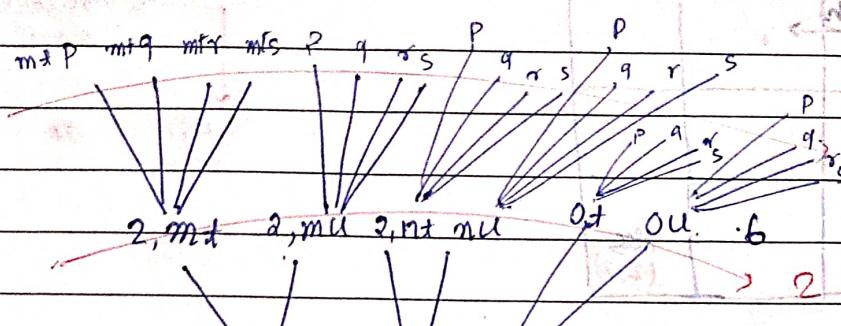
$5 \rightarrow mno$

$6 \rightarrow pqrs$

$7 \rightarrow tu$

$8 \rightarrow vwx$

$9 \rightarrow yz$



way up

Return type

[mno, pqrs, tu]

Ans: 567 → {mno, pqrs, tu} = {{5, 6, 7} → {1, 2, 3}}

faith :-

$5 \rightarrow mno$

$6 \rightarrow pqrs$

$7 \rightarrow tu$

$5\#6\#7$ → [p_t, p_u, q_t, q_u, r_t, r_u, s_t, s_u]

$5\#6\#7$

[m_p, m_q, m_r, m_s, m_t, m_u, m_v, m_w, m_x, m_y, m_z]

Sfr → "567"

$5 \rightarrow mno$

$6 \rightarrow pqrs$

$7 \rightarrow tu$

{2, 4, 6, 8} → [t, u] → tu

[t, u]

i, 6

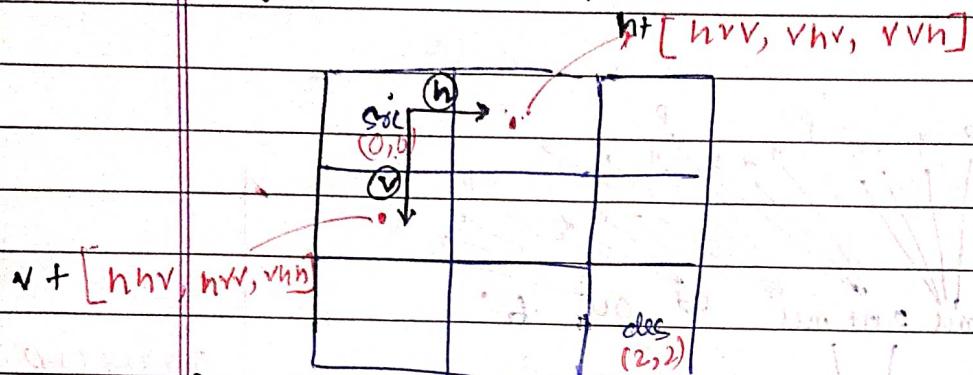
[p_t, p_u, q_t, q_u, r_t, r_u, s_t, s_u] → pqrs

[p_t, p_u, q_t, q_u, r_t, r_u, s_t, s_u]

[m_p, m_q, m_r, m_s, m_t, m_u, m_v, m_w, m_x, m_y, m_z]

0, 5

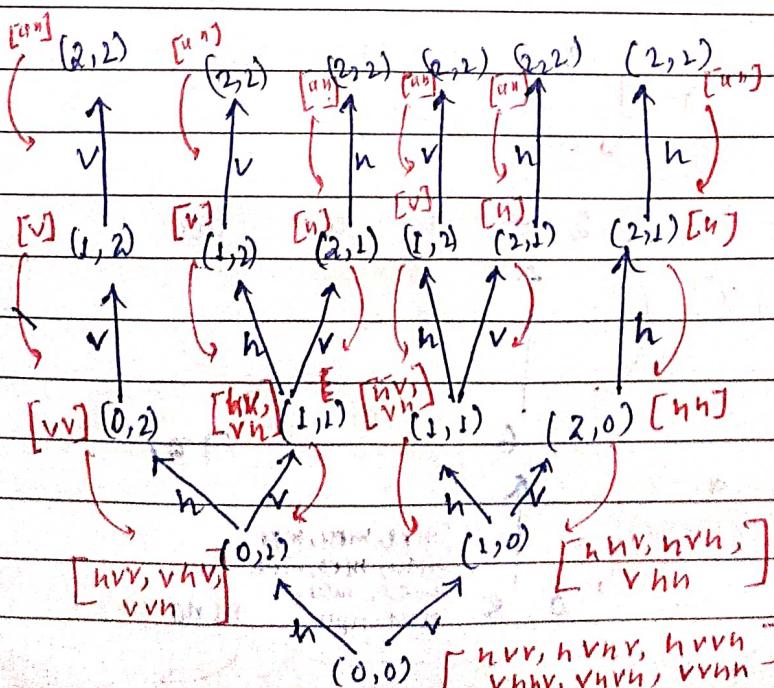
[n_p, n_q, n_r, n_s, n_t, n_u, n_v, n_w, n_x, n_y, n_z] → mno

Programming Get Maze pathsFaith

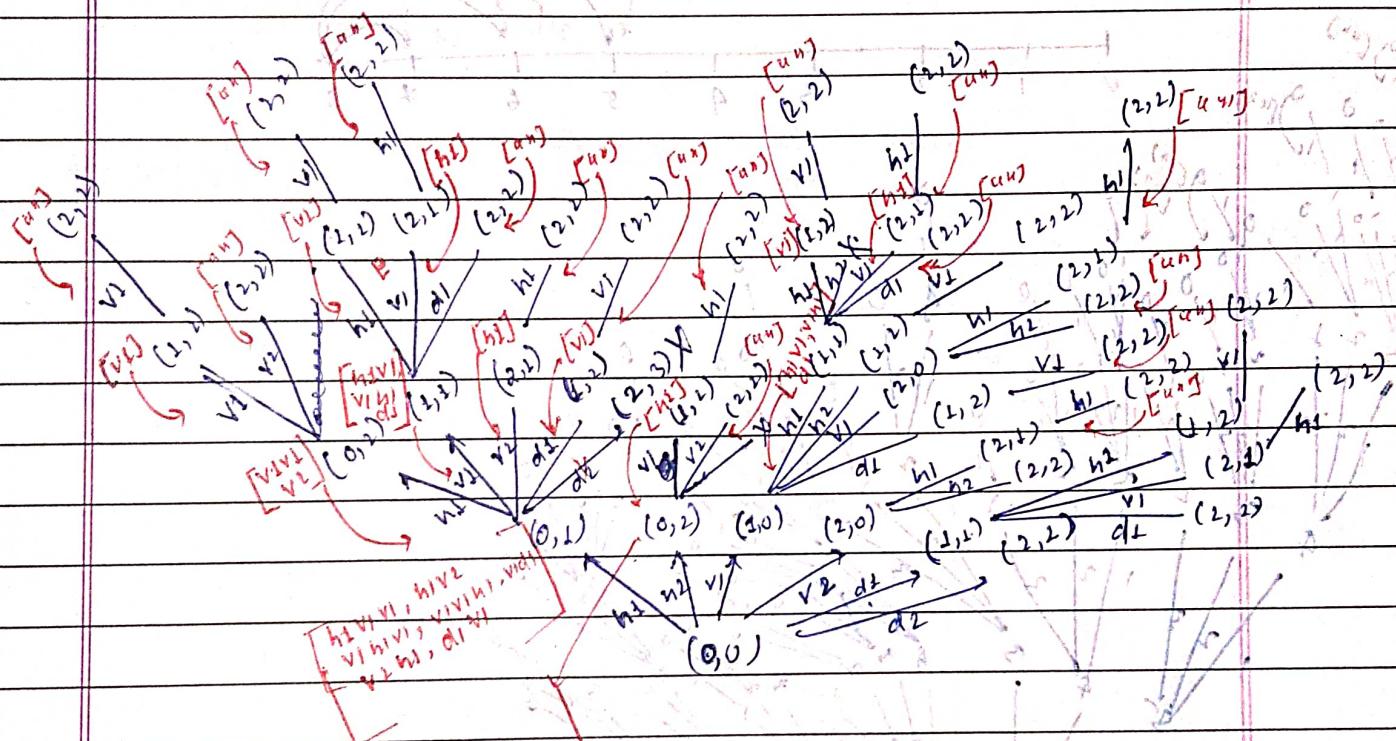
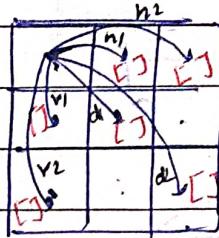
- ① $(0,1) \rightarrow (2,2)$, saare path ek arraylist me store karke
de do

~~[hvv, vhv, vvh]~~
baad me us arraylist me "h" add kardo
[hhvv, hvhv, hvvh]

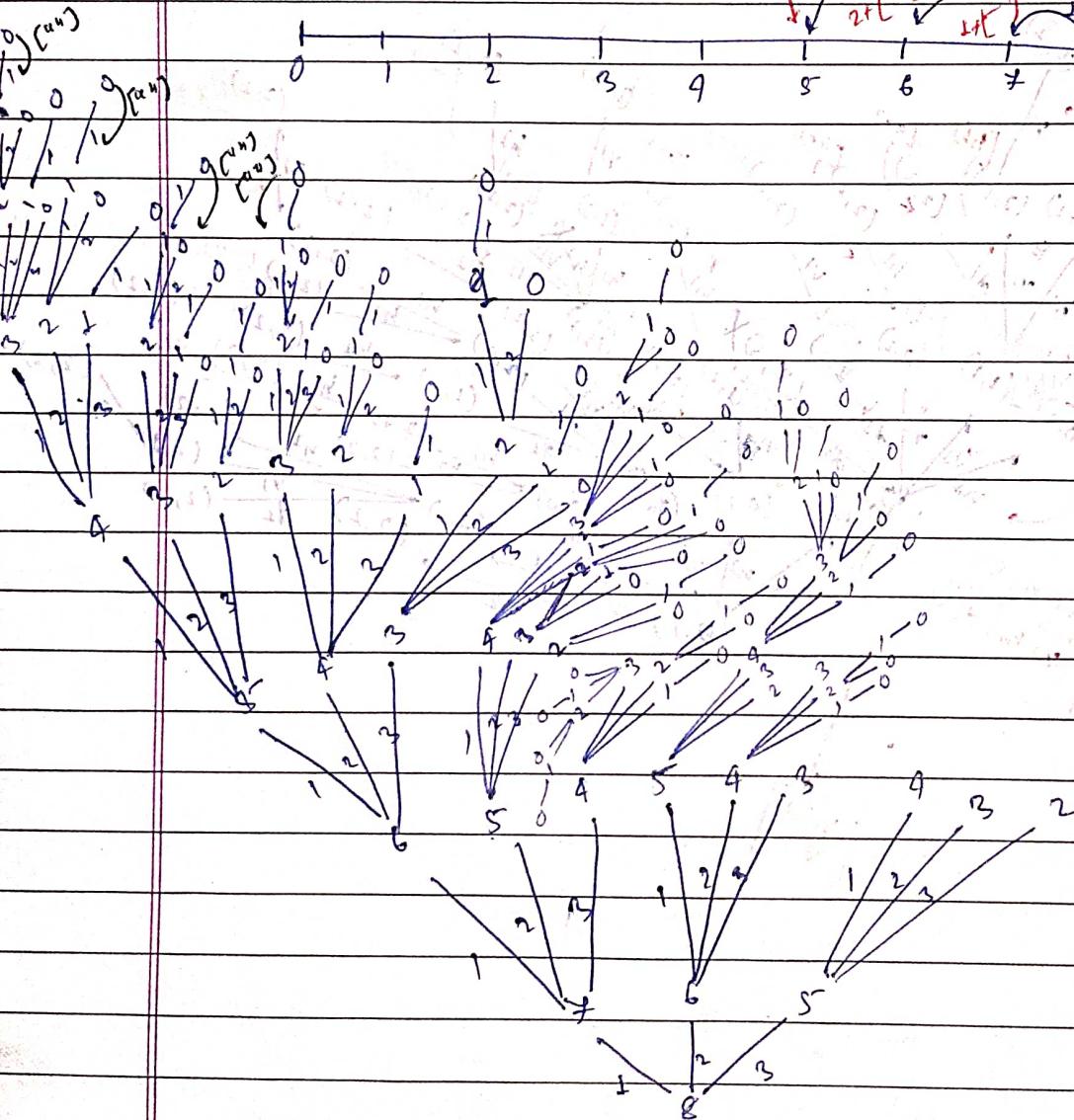
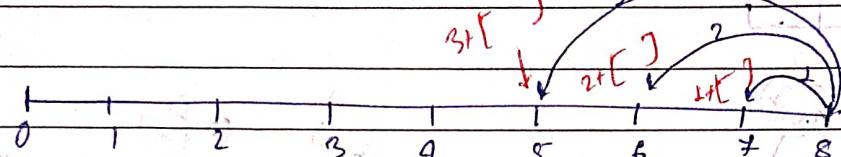
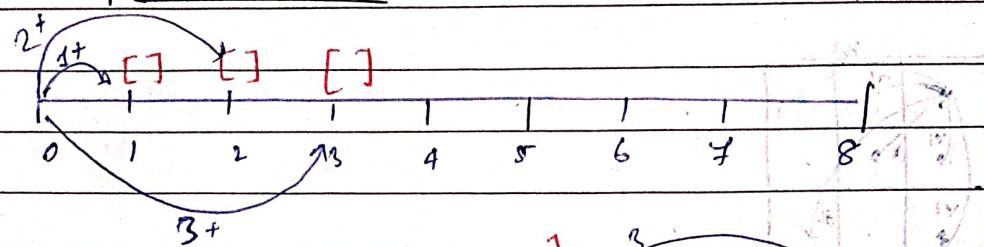
- ② $(1,0) \rightarrow (2,2)$ saare path ek arraylist me store karke do
[nhv, nhv, vuh]
baad me us arraylist me "v" add kardo
[vhn, vhv, vvh]

tree diagram:

Programs Get Maze Path with Jumps



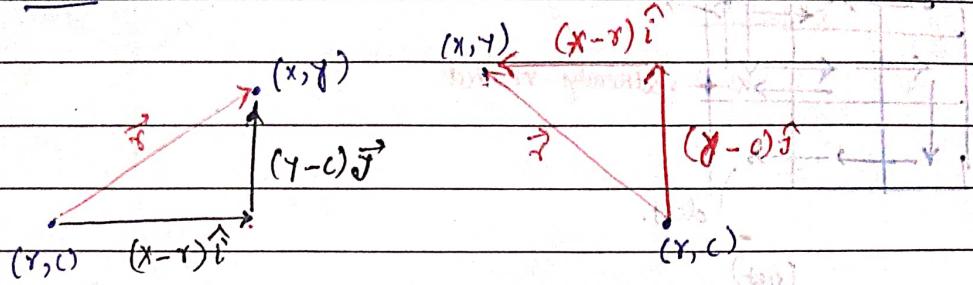
Problem 8 Get stairs path (allowed 1 step, 2 step, 3 step)





* * * FLOOD FILL Algorithm

Vector



4-direction

Up (-1, 0)

left (0, -1)

Right (0, 1)

down (1, 0)

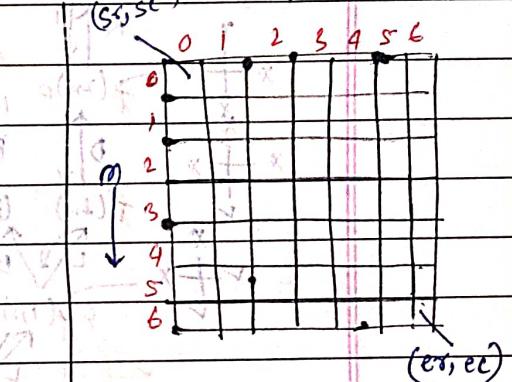
[extraction array]
dire

0	0	-1
1	1	0
2	0	1
3	-1	0

(direction string) → Array
dire8

(S, sc)

m →



$$(x, y) = (r, c) + (\hat{i}, \hat{j})$$

$$x = sr + \text{dire}[d][0]$$

$$y = sc + \text{dire}[d][1]$$

$$(8, 6) + (0, -1) \rightarrow (8, 5) \quad \text{(left dire)}$$

$$(8, 6) + (1, 0) \rightarrow (9, 6) \quad \text{(down dire)}$$

$$(8, 6) + (0, 1) \rightarrow (8, 7) \quad \text{(Right dire)}$$

$$(8, 6) + (-1, 0) \rightarrow (7, 6) \quad \text{(up dire)}$$

pseudo code (for diren find)

```
for (int d = 0; d < dire.length; d++) {
```

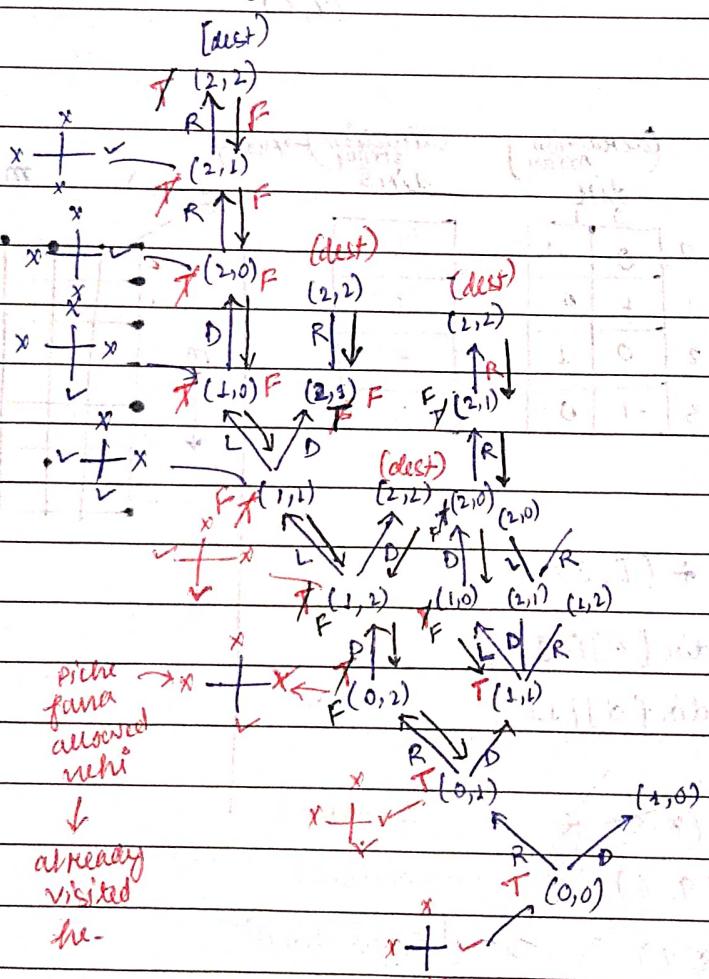
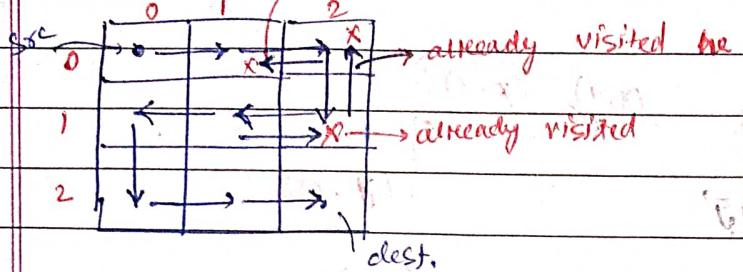
```
    int r = sr + dire[d][0];
```

```
    int c = sc + dire[d][1];
```

```
}
```

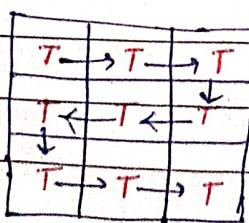
dry run

Piche fana allowed (at already visited path ne)



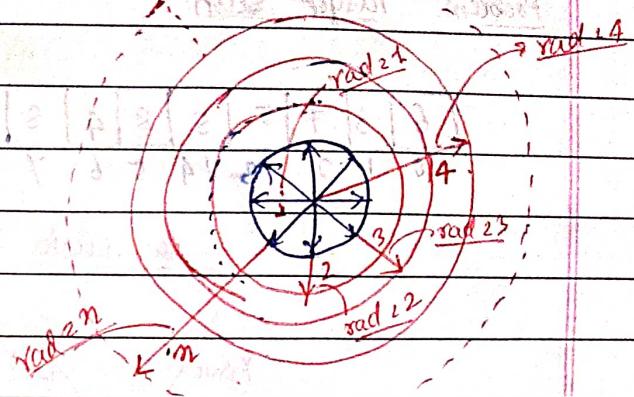
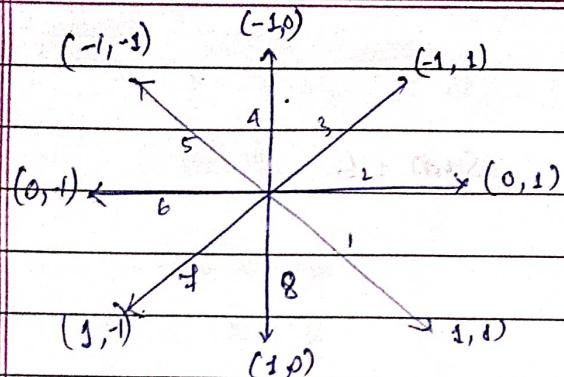
Upar wale dry run se mujhe ek baad samajh aya he

- ① fab bhi hum first time kisi khesh pe visit karange toh usko visited mark kar denge. (boolean true store karne ka rakh denge)



humne isse ye maloom chalta he ki
true wall cell pe fana allowed rakh
ke.

- ② Aur Jati samai (Back track ke time) visited cell (~~true~~ true)
khesh ko unvisited karne khi hamari jismmedari ke
(boolean false store karne ka rakh denge)



for rad 21 \rightarrow magnitude

$x = sr + 1 * \text{dire}[d][0]$ \rightarrow direction

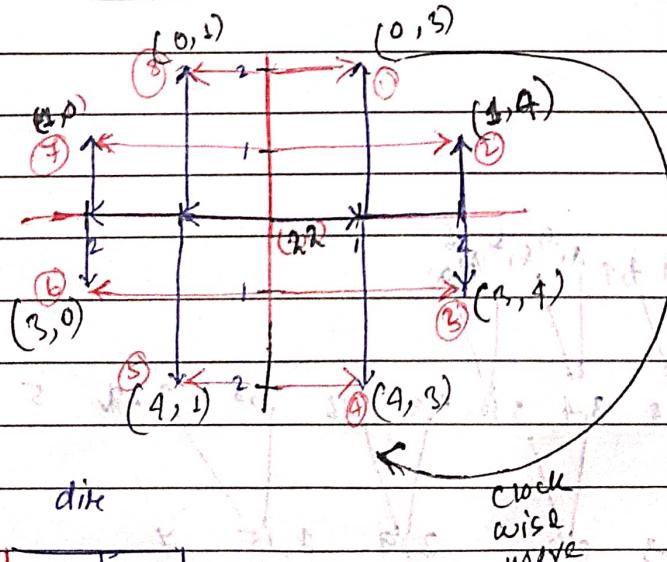
$\gamma_2 \text{se} + 1 \# \text{dim}[d][1]$

for $\text{rad} = n$

$$X = sr + rad \times \dim [d] [o]$$

$y_2 = \text{sc} + \text{rad} * \text{dir}(\text{adj}[j])$

Program 8 Knights Tour (8 possible move)



0	-2	1
1	-1	R
2	1	3
3	2	4
4	2	-1
5	1	-2
6	-1	-2
7	-2	-1



Date 3/07/2021

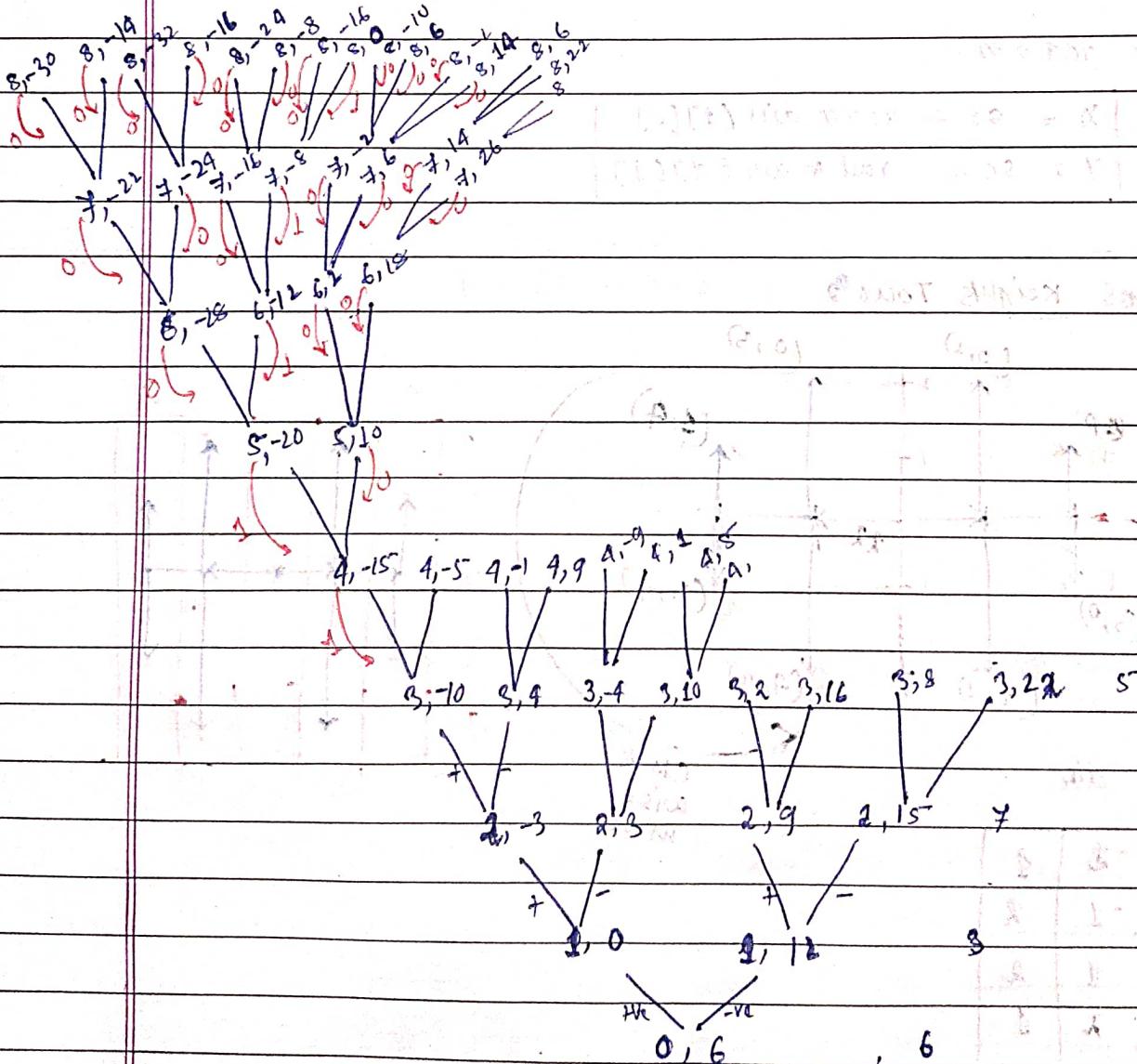
Date / /
Page

Problem: Target Sum

6	3	7	5	5	8	4	8
0	1	2	3	4	5	6	7

Sum = 6

two choice
(+ve) (-ve)

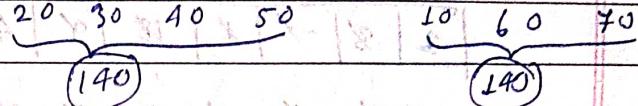


problem Equivalent Set

$$S^2 \ni 10, 20, 30, 40, 50, 60, 70 \quad n=140$$

Set A Set B

$$\left\{ \begin{array}{l} A \cup B = S \\ A \cap B = \emptyset \\ \Sigma A = \Sigma B \end{array} \right.$$



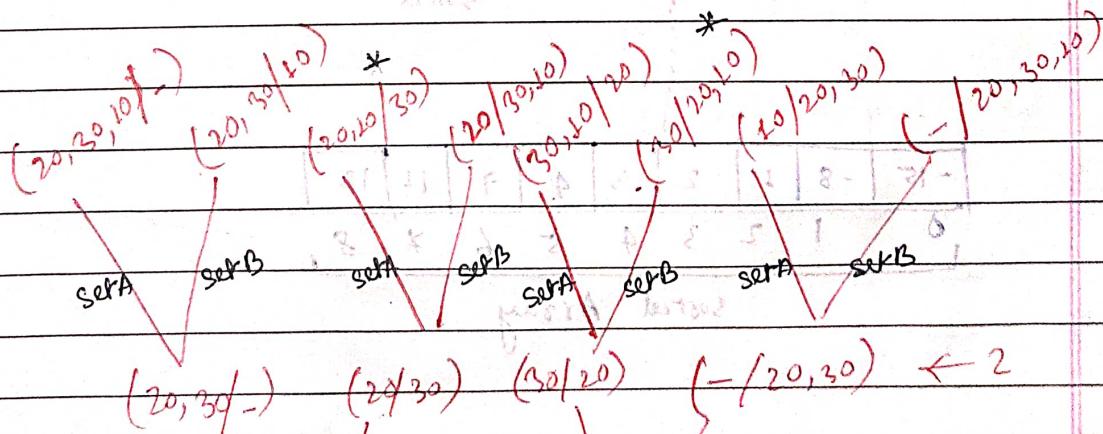
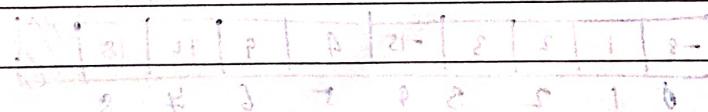
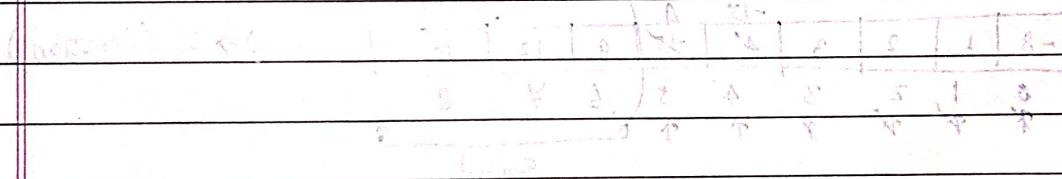
$$\text{print: } -20, 30, 40, 50 = 10, 60, 70$$

Algorithm

Faith \rightarrow ① Array ka har ek element Set A ka part banega ya

Set B ka part banega.

② Agar set A ka part banta hu toh mujhe sum1 me add koro., Agar set B \rightarrow sum2 me add koro.



[20, 30, 10]
0 1 2

(-/-) $\leftarrow 0$
SetA SetB
Index