Print all path with minimum Jumps: →

arr →

0	1	2	3	4	5	6	7	8	9
3	3	0	2	1	2	4	2	0	0

4	4	∞	2	2	2	1	1	∞	0

↓ marker

arr[i] → no. of Jumps allowed from ith index

0 to 9 → Min Jumps

Smallest problem → 9 to 9 = 0
↳ no. of jumps

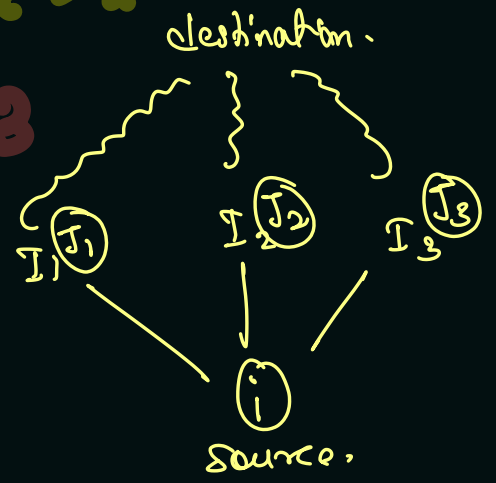
9 to 9 → no. of paths
↳ 1
i.e. don't move.

Jump vs path

3 → 2 → 2 → 2

3 → 2 → 1 → 3

Size vs Jump Count



i to destination →

$= \min(J_1, J_2, J_3) + 1$ → from source to intermediate

Result from arr[0] to arr[9] we can Reach in 4 Jumps

path ??

	0	1	2	3	4	5	6	7	8	9
arr →	3	3	0	2	1	2	4	2	0	0

4	4	∞	3	3	2	1	1	∞	0

→ Prodgi...

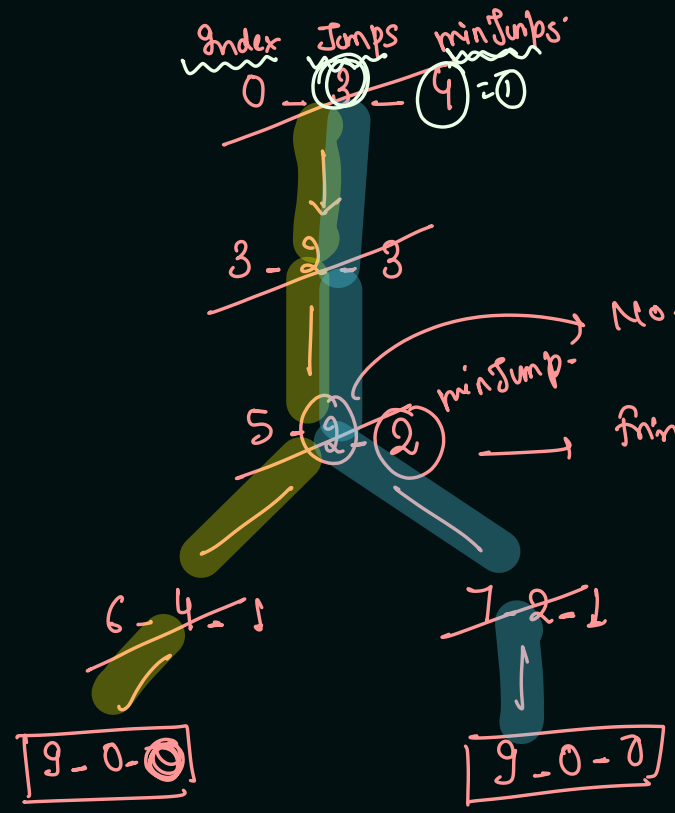
0 → 3 → 5 → 6 → 9.

3 → 2 → 1 → 3.

0 → 3 → 5 → 7 → 9.

3 → 2 → 2 → 2.

Reverse Engineering



No. of jump allowed, help to find reachable index

find minJumps - 1 from Reachable index

DP.

① Storage

② Assign meaning to cell

③ Fill Prequisite

↑

Analyse Smaller

→ BFS.

① Get + Remove proble

② make and direct

③ work to fill DP

④ Add unvisited nbs.

④ Travel & Fill


```java
Queue<MJHelper> que = new LinkedList<>();
que.add(new MJHelper(0, arr[0], dp[0], "0"));

while(que.size() > 0) {
    MJHelper rem = que.remove();
    if(rem.indx == n - 1) {
        System.out.println(rem.psf + " .");
        continue;
    }
    for(int jump = 1; jump <= rem.jumps && jump + rem.indx < n; jump++) {
        if(dp[rem.indx + jump] == rem.minJumps - 1) {
            int nindx = rem.indx + jump;
            que.add(new MJHelper(nindx, arr[nindx], rem.minJumps - 1, rem.psf + " -> "+ nindx))
        }
    }
}
```

indx jmp minJmp, psf

0, 3, 4, 0

3, 2, 3, 0 → 3

5, 2, 2, 0 → 3 → 5

6, 4, 1, 0 → 3 → 5 → 6

7, 2, 1, 0 → 3 → 5 → 7

9, 0, 0, 0 → 3 → 5 → 6 → 9

9, 0, 0, 0 → 3 → 5 → 7 → 9

0 → 3 → 5 → 6 → 9

0 → 3 → 5 → 7 → 9

Print all path for Min cost

	0	1	2	3	4	5
0	0	1	4	2	8	2
1	4	3	6	5	0	4
2	1	2	4	1	4	6
3	2	0	7	3	2	2
4	3	1	5	9	2	4
5	2	7	0	8	5	1

	0	1	2	3	4	5
0	23	23	24	20	21	19
1	24	22	23	18	13	17
2	20	19	17	18	13	13
3	21	19	19	12	9	7
4	23	20	19	16	7	8
5	23	21	14	14	6	1

Horizontal

vertical

with two moves ie. H, & V,
what is the min cost to reach
at Bottom Right from top
left?

H V V H H V H V H V

H V V H H V H H V V

Point All path for max Gold :

Goldmine → D1, D2, D3

x, y → $x-1, y+1$; $x, y+1$; $x+1, y+1$

last Colun

first Row

	0	1	2	3	4	5
0	0	1	4	2	8	2
1	4	3	6	5	0	4
2	1	2	4	1	4	6
3	2	0	7	3	2	2
4	3	1	5	9	2	4
5	2	7	0	8	5	1

Last

	0	1	2	3	4	5
0	26	24	21	14	12	2
1 (83)	31	26	23	17	6	4
2 (83)	28	27	21	11	10	6
3	29	25	25	13	8	2
4 (33)	33	26	23	18	6	4
5	32	30	18	17	9	1

(33) (4)

D3 → D1 → D2 → D3 → D1

Direction of Mining → left to Right

Direction of solving problem → Right to left

Last Column

→ Same as min

$$dp[r][c] = mine[r][c]$$

First Row

$$dp[r][c] \rightarrow \left.\begin{matrix} D2 \\ D3 \end{matrix}\right] \underbrace{\text{Max}}_{\text{from DP}} + mine[r][c]$$

Last Row

$$dp[r][c] \rightarrow \left.\begin{matrix} D1 \\ D_2 \end{matrix}\right] \underbrace{\text{max}}_{\text{from DP}} + mine[r][c]$$

Middle.

$$dp[r][c] \rightarrow \left.\begin{matrix} D1 \\ D_2 \\ D3 \end{matrix}\right] \text{max from DP} + mine[r][c]$$

	0	1	2	3	4	5
0	26	24	21	14	12	2
1	31	26	23	17	6	4
2	28	27	21	11	10	6
3	29	25	25	13	8	2
4	33	26	23	18	6	4
5	32	30	18	17	9	4

if $\underbrace{c == 5}$

Destination is arrived

4 D3 D1 D2 D3 D1

① Travel in zero th column and add max coordinate in queue of pairs.

② Then solve BFS

x, y, psf
i, j, "i"

~~4, 0, "4"~~

~~5, 1, "4 D3"~~

~~4, 2, 4 D3 D1~~

~~4, 3, "4 D3 D1 D2"~~

~~5, 4, 4 D3 D1 D2 D3~~

~~4, 5, 4 D3 D1 D2 D3 D1~~

5, 5, 4 D3 D1 D2 D3 D2

$x \rightarrow$	-1	0	1
$y \rightarrow$	1	1	1

x, y

$D1$ — $x-1, y+1$

$D2$ — $x, y+1$

$D3$ — $x+1, y+1$

$D = 1$

$D = 2$

$D = 3$