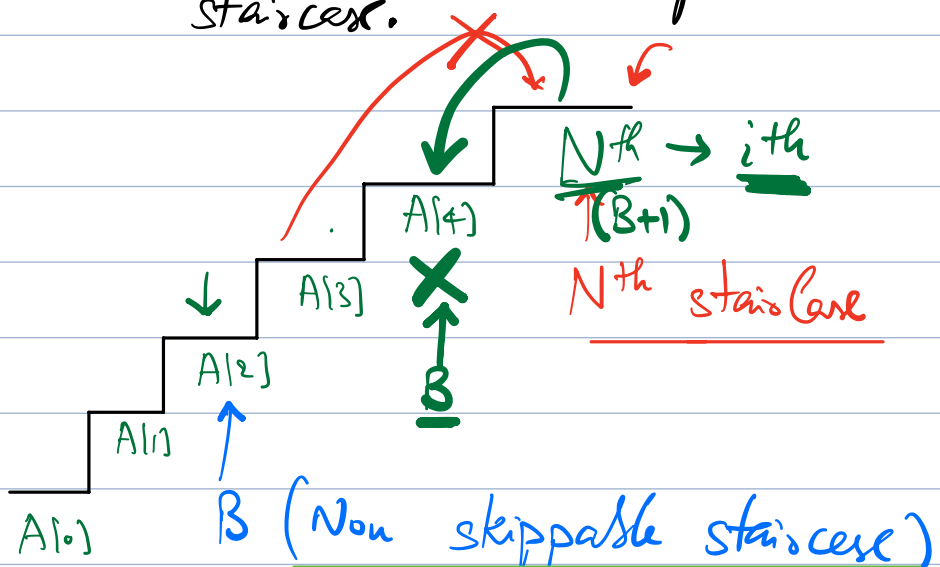
# Min Cost with Non Skippable Staircase.

Contest Question Score.

Array of length N

A[i] represents the cost of ith staircase.




$$A = [\underset{0}{4}, \underset{1}{8}, \underset{2}{6}, \underset{3}{5}, \underset{4}{1}, \underset{5}{10}, \underset{6}{2}]$$

1 Element Choice

Step size 1

$$minCost(i) = \min\left(minCost(i-1) + A[i-1],\ minCost(i-2) + A[i-2]\right)$$

```
if (i == B+1) {
    minCost(i) = minCost(i-1) + A[i-1];
}
```

$$DP[i] = A[i] + \min(DP[i-1], DP[i-2])$$

Checked ✓

$$A = [\underset{0}{4}, \underset{1}{8}, \underset{2}{6}, \underset{3}{5}, \underset{4}{1}, \underset{5}{10}, \underset{6}{2}] \quad 7^{th}$$

$$DP[8] = [\underset{0}{0}, \underset{1}{0}, \underset{2}{4}, \underset{3}{8}, \underset{4}{10}, \underset{5}{11}, \underset{6}{21}, \underset{7}{21}]$$

(4 if B == 0)

(19)

```
for (i = 2; i <= N; i++) {
    if (i == B+1) {
        DP[i] = A[i-1] + DP[i-1];
    } else {
        DP[i] = min(A[i-1] + DP[i-1],
                    A[i-2] + DP[i-2]);
```

}

}

B = 2

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1, & 3, & 2, & 4, & 5 \end{bmatrix}$$

---

# Distance o/p Nearest cell

Matrix A { size N x M. < 0, 1

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

⇒

| | | | |
|---|---|---|---|
| 1 | 2 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |

$|x_1 - x_2| + |y_1 - y_2|$ } Distance

↓

↑

← (i, j) →

↓

(2, 3) (0, 1)

$|2-0| + |3-1|$

$= 2+2=4$

App 1 $\Rightarrow$ $\forall i,j$ where $M[i][j] == 0$

bfs $(i,j)$ $\Rightarrow$ Untill you find the 1st one.

T.C. $= O(V+E) \times V^2$

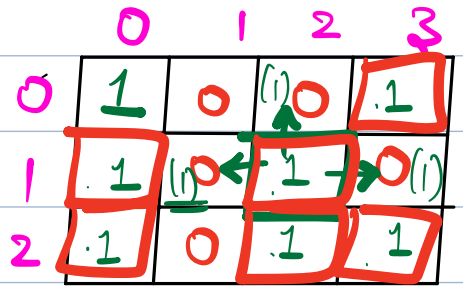
$O(N^2 + 4N^2)$

$= O(N^2) \times (N^2)$

$= O(N^4)$

$\therefore N \leq ((10)^3)^4$

$= 10^{12} X$




$\forall_{1^s} \Rightarrow O(NM) \times O(NM)$

$= O(N^2M^2)$

★ All 1s are identical.

⇓

Start a single BFS with multiple source nodes.

```
vst[N][M] = {false};
Queue< Pair< Pair<int, int>, int> > q
                 i,  j
for (i=0; i<N; i++) {
    for (j=0; j<M; j++) {
```

```
if (M[i][j] == 1) {
    q.enqueue(<<i, j>, 0>);
    vst[i][j] = True;
}
}
}
```

```
while (!q.isEmpty()) {
    x = q.dequeue();
    i = x.first.first;
    j = x.first.second;
    d = x.second;
    for (k=0; k<row.size(); k++) {
        if (check(i+row[k], j+col[k]) {
            if (vst[i+row[k]][j+col[k]]==false) {
                ans[i+row[k]][j+col[k]] = d+1;
                q.enqueue(<<i+row[k], j+col[k]>, d+1>);
            }
```

{-1, 1, 0, 0}
{0, 0, 1, -1}

R ⇒ Residence (0)
H ⇒ Hospital (1)

Distance of every residence to nearest hosp.

| | | | |
|---|---|---|---|
| $R_1$ | $R_2$ | $R_3$ | $H_1$ |
| $R_4$ | $R_5$ | $H_2$ | $H_3$ |
| $R_6$ | $H_4$ | $H_5$ | $R_7$ |

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

~~$\langle H_1, 0\rangle$~~, ~~$\langle H_2, 0\rangle$~~, ~~$\langle H_3, 0\rangle$~~, ~~$\langle H_4, 0\rangle$~~, ~~$\langle H_5, 0\rangle$~~, ~~$\langle R_3, 1\rangle$~~

~~$\langle R_5, 1\rangle$~~, ~~$\langle R_7, 1\rangle$~~, ~~$\langle R_6, 1\rangle$~~, ~~$\langle R_2, 2\rangle$~~, ~~$\langle R_4, 2\rangle$~~, ~~$\langle R_1, 3\rangle$~~

~~$\langle H_1, 0\rangle$~~ ~~$\langle H_2, 0\rangle$~~, ~~$\langle H_3, 0\rangle$~~

$\langle H_4, 0\rangle$, $\langle R_3, 1\rangle$

T.C. = $O(N \times M \times 4)$

$= O(N \times M)$

Merge Test Case

$A = 1$ $B[10000000]$

$A = \Rightarrow [(A[0] \times B.), (\ ), (\ ) \ldots]$

0, 0, 1, 2,

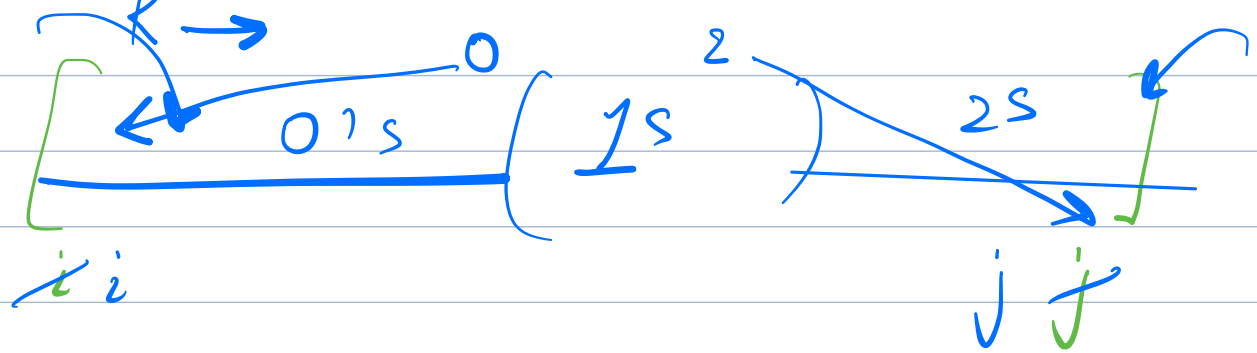
B, 1, 2

0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2

count 0 =

1 =

2 =


0
0's
1's
2
2's
i i
j j