

```

from collections import deque
directions = [[0, 1], [1, 0], [-1, 0], [0, -1]]
def bfs(w, h, a, x, y, c, d):
    a[c][d]=" "
    q = deque()
    q.append((x,y,1,0))
    q.append((x, y, -1, 0))
    q.append((x, y, 0, 1))
    q.append((x, y, 0, -1))
    inq = set()
    inq.add((x, y))
    step=0
    while q:
        for _ in range(len(q)):
            s, t,ds,dt=q.popleft()
            if s==c and t==d:
                a[c][d]="X"
                return step
            for dx, dy in directions:
                if dx!=ds and dy!=dt:
                    for k in range(1,max(w,h)+2):
                        nx, ny = s + k*dx, t + k*dy
                        if 0 <= nx <= h + 1 and 0 <= ny <= w + 1 and a[nx][ny] ==
" " and (nx, ny) not in inq:
                            q.append((nx,ny,dx,dy))
                            inq.add((nx, ny))
                        elif 0 <= nx <= h + 1 and 0 <= ny <= w + 1 and a[nx][ny]
!= " ":
                            break
                    step+=1
            return "No"
o_1 = 0
while True:
    o_1 += 1
    w, h = map(int, input().split())
    if w == 0 and h == 0:
        break
    a = [[" "] * (w + 2)]
    a += [[" "] + list(input()) + [" "] for _ in range(h)]
    a += [[" "] * (w + 2)]
    print(f"Board #{o_1}:")
    o_2 = 0
    while True:
        o_2 += 1
        y, x, d, c = map(int, input().split())
        if x == 0 and y == 0 and c == 0 and d == 0:
            break
        l = bfs(w, h, a, x, y, c, d)
        if l == "No":
            print(f"Pair {o_2}: impossible.")
        else:
            print(f"Pair {o_2}: {l} segments.")
    print()

```

