

40. Destination City

You are given the array `paths`, where `paths[i] = [cityAi, cityBi]` means there exists a direct path going from `cityAi` to `cityBi`. Return the destination city, that is, the city without any path outgoing to another city. It is guaranteed that the graph of paths forms a line without any loop, therefore, there will be exactly one destination city. Example 1: Input: `paths = [["London","New York"],["New York","Lima"],["Lima","Sao Paulo"]]` Output: `"Sao Paulo"` Explanation: Starting at `"London"` city you will reach `"Sao Paulo"` city which is the destination city. Your trip consist of: `"London" -> "New York" -> "Lima" -> "Sao Paulo"`. Example 2: Input: `paths = [["B","C"],["D","B"],["C","A"]]` Output: `"A"` Explanation: All possible trips are: `"D" -> "B" -> "C" -> "A"`. `"B" -> "C" -> "A"`. `"C" -> "A"`. `"A"`. Clearly the destination city is `"A"`. Example 3: Input: `paths = [["A","Z"]]` Output: `"Z"`

PROGRAM:

```
def destCity(paths):
```

```
    destinations = set()
```

```
    for path in paths:
```

```
        destinations.add(path[1])
```

```
    for path in paths:
```

```
        if path[0] not in destinations:
```

```
            return path[0]
```

```
    return None
```

```
paths1 = [["London","New York"],["New York","Lima"],["Lima","Sao Paulo"]]
```

```
print(destCity(paths1))
```

```
paths2 = [["B","C"],["D","B"],["C","A"]]
```

```
print(destCity(paths2))
```

```
paths3 = [["A","Z"]]
```

```
print(destCity(paths3))
```

OUTPUT:

```
London
D
A
```

TIME COMPLEXITY: $O(n)$