Java

```java
class Solution {
    public int minimumDistance(int n, List<List<Integer>> edges, int s, int[] marked) {

    }
}
--------------------------------------------------------------------
```

JavaScript

```javascript
/**
 * @param {number} n
 * @param {number[][]} edges
 * @param {number} s
 * @param {number[]} marked
 * @return {number}
 */
var minimumDistance = function(n, edges, s, marked) {

};
--------------------------------------------------------------------
```

TypeScript

```typescript
function minimumDistance(n: number, edges: number[][], s: number, marked: number[]): number {

};
--------------------------------------------------------------------
```

C++

```
class Solution {
public:
    int minimumDistance(int n, vector<vector<int>>& edges, int s, vector<int>& marked) {

    }
};
```

C#

```
public class Solution {
    public int MinimumDistance(int n, IList<IList<int>> edges, int s, int[] marked) {

    }
}
```

Kotlin

```
class Solution {
    fun minimumDistance(n: Int, edges: List<List<Int>>, s: Int, marked: IntArray): Int {

    }
}
```

Go

```
func minimumDistance(n int, edges [][]int, s int, marked []int) int {

}
```