Java

```java
class Solution {
    public int minimumCost(int n, int[][] connections) {

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

JavaScript

```javascript
/**
 * @param {number} n
 * @param {number[][]} connections
 * @return {number}
 */
var minimumCost = function(n, connections) {

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

TypeScript

```typescript
function minimumCost(n: number, connections: number[][]): number {

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

C++

```cpp
class Solution {
```

```
public:
    int minimumCost(int n, vector<vector<int>>& connections) {

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

C#

```
public class Solution {
    public int MinimumCost(int n, int[][] connections) {

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

Kotlin

```
class Solution {
    fun minimumCost(n: Int, connections: Array<IntArray>): Int {

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