Java

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

    }
}
```

JavaScript

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

};
```

TypeScript

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

};
```

C++

```cpp
class Solution {
public:
    vector<int> minimumCost(int n, vector<vector<int>>& edges, vector<vector<int>>& query) {

    }
};
```

C#

```cs
public class Solution {
    public int[] MinimumCost(int n, int[][] edges, int[][] query) {

    }
}
```

Kotlin

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

    }
}
```

Go

```go
func minimumCost(n int, edges [][]int, query [][]int) []int {

}
```
