Java

```java
class Solution {
    public long[] minCost(int n, int[][] roads, int[] appleCost, int k) {

    }
}
```

JavaScript

```javascript
/**
 * @param {number} n
 * @param {number[][]} roads
 * @param {number[]} appleCost
 * @param {number} k
 * @return {number[]}
 */
var minCost = function(n, roads, appleCost, k) {

};
```

TypeScript

```typescript
function minCost(n: number, roads: number[][], appleCost: number[], k: number): number[] {

};
```

C++

```
class Solution {
public:
    vector<long long> minCost(int n, vector<vector<int>>& roads, vector<int>& appleCost, int k) {

    }
};
```

C#

```
public class Solution {
    public long[] MinCost(int n, int[][] roads, int[] appleCost, int k) {

    }
}
```

Kotlin

```
class Solution {
    fun minCost(n: Int, roads: Array<IntArray>, appleCost: IntArray, k: Int): LongArray {

    }
}
```

Go

```
func minCost(n int, roads [][]int, appleCost []int, k int) []int64 {

}
```