

Java

```
class Solution {  
    public int minCost(int[] startPos, int[] homePos, int[] rowCosts, int[] colCosts) {  
  
    }  
}
```

JavaScript

```
/**  
 * @param {number[]} startPos  
 * @param {number[]} homePos  
 * @param {number[]} rowCosts  
 * @param {number[]} colCosts  
 * @return {number}  
 */  
var minCost = function(startPos, homePos, rowCosts, colCosts) {  
  
};
```

TypeScript

```
function minCost(startPos: number[], homePos: number[], rowCosts: number[], colCosts: number[]): number {  
  
};
```

C++

```
class Solution {
public:
    int minCost(vector<int>& startPos, vector<int>& homePos, vector<int>& rowCosts, vector<int>& colCosts) {

    }
};
```

C#

```
public class Solution {
    public int MinCost(int[] startPos, int[] homePos, int[] rowCosts, int[] colCosts) {

    }
}
```

Kotlin

```
class Solution {
    fun minCost(startPos: IntArray, homePos: IntArray, rowCosts: IntArray, colCosts: IntArray): Int {

    }
}
```

Go

```
func minCost(startPos []int, homePos []int, rowCosts []int, colCosts []int) int {

}
```

