

*// Pior caso quando todas as diagonais estão contidas no vetor*

```
unsigned int* better_search_and_count(int** matriz, int* vetor, int m, int z) {  
    unsigned int* con = calloc(z, sizeof(unsigned int));  
  
    for (int k = 0; k < z; k++) { // Repete z vezes  
        int element = vetor[k];  
  
        for (int i = 0; i < m; i++) { // Repete m * z vezes  
            if (element < matriz[i][i]) { // c  
                for (int j = i - 1; j >= 0; j--)  
                    if (matriz[i][j] == element) con[k]++;  
  
            } else if (element > matriz[i][i]) { // c  
                for (int j = i + 1; j < m; j++)  
                    if (matriz[i][j] == element) con[k]++;  
  
            } else {  
                for (int j = 0; j < m; j++) // Repete m * m * z  
                    if (matriz[i][j] == element) con[k]++; // c + a  
            }  
        }  
    }  
  
    return con;  
}
```

```
/*  
f(n) = m*z*(c + c + m(c + a))  
f(n) = m*z*(2c + m(c + a))  
f(n) = m*z*2c + m2*z*(c + a)
```

*Equação Fechada:  $f(n) = m \cdot z \cdot [2c + (c + a)/m]$*

*Big O notation:  $O(m \cdot z)$*

```
*/
```