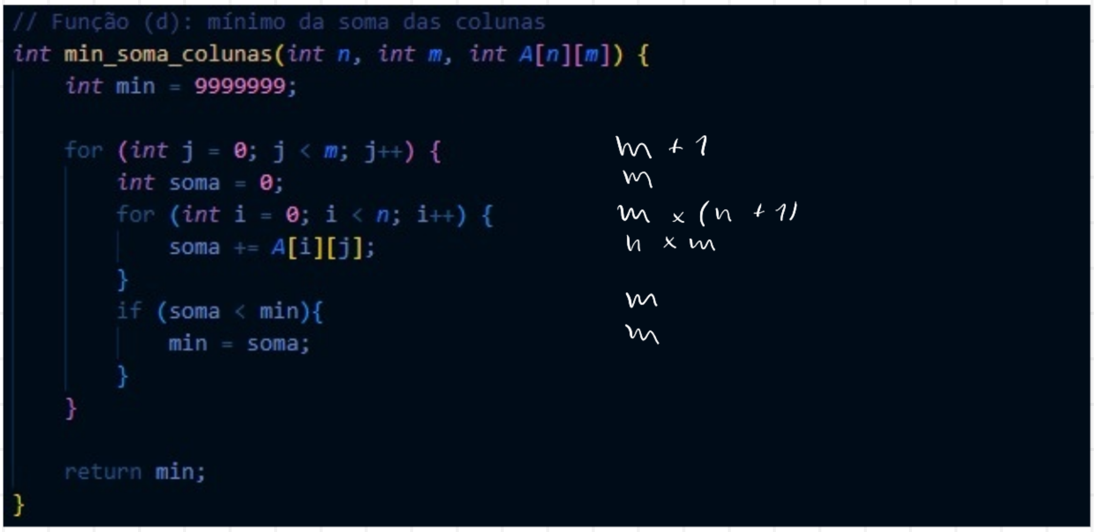
Heverton Souza

Complexidades

1 -

d)

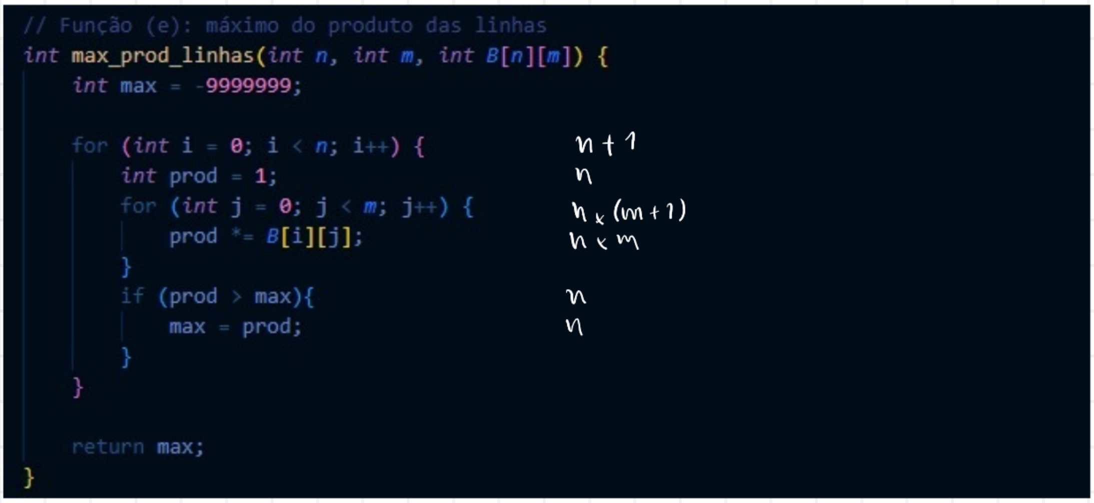


```
// Função (d): mínimo da soma das colunas
int min_soma_colunas(int n, int m, int A[n][m]) {
    int min = 9999999;

    for (int j = 0; j < m; j++) {                 m + 1
        int soma = 0;                             m
        for (int i = 0; i < n; i++) {             m x (n + 1)
            soma += A[i][j];                      n x m
        }
        if (soma < min){                          m
            min = soma;                           m
        }
    }

    return min;
}
```

$T(n,m) = (m+1) + m + (m.(n+1)) + (n.m) + m + m = m+1+m+mn+m+nm+m+m =$

$= 2mn + 5m + 1 \longrightarrow O(nm)$

e)

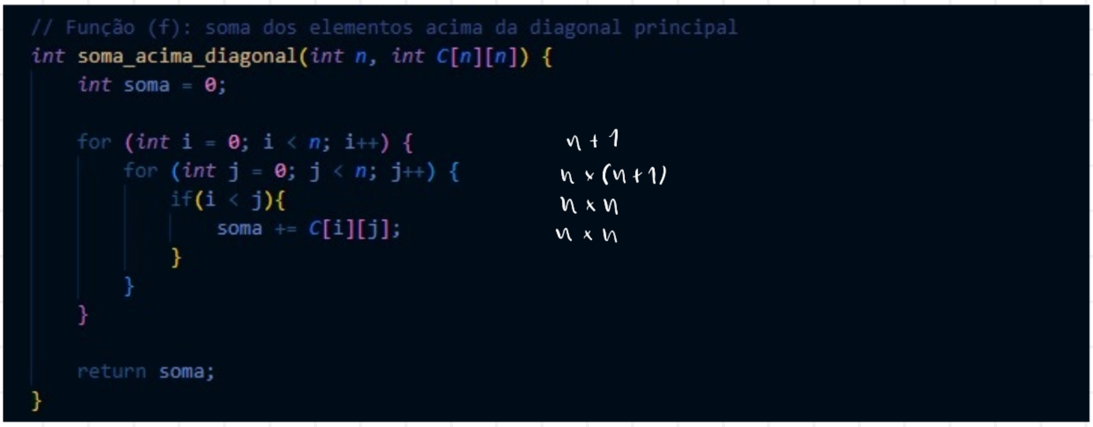


```
// Função (e): máximo do produto das linhas
int max_prod_linhas(int n, int m, int B[n][m]) {
    int max = -9999999;

    for (int i = 0; i < n; i++) {                n + 1
        int prod = 1;                            n
        for (int j = 0; j < m; j++) {            n x (m + 1)
            prod *= B[i][j];                     n x m
        }
        if (prod > max){                         n
            max = prod;                          n
        }
    }

    return max;
}
```

$T(n,m) = (n+1) + n + (n.(m+1)) + (n.m) + n + n = n+1+n+mn+n+nm+n+n =$

$= 2nm + 5n + 1 \longrightarrow O(nm)$

f)



```
// Função (f): soma dos elementos acima da diagonal principal
int soma_acima_diagonal(int n, int C[n][n]) {
    int soma = 0;

    for (int i = 0; i < n; i++) {               n + 1
        for (int j = 0; j < n; j++) {           n x (n + 1)
            if(i < j){                          n x n
                soma += C[i][j];                n x n
            }
        }
    }

    return soma;
}
```

$T(n) = (n+1) + (n.(n+1)) + (n.n) + (n.n) = n+1+n^2+n+n^2+n^2$

$= 3n^2 + 2n + 1 \longrightarrow O(n^2)$