SelectionSort(arr, n)

```
for i ← 1 to n-1
    min ← i                          O(1)
    for j ← i+1 to n
        if A[j] < A[min] then        O(1)
            min ← j                  O(1)
    end_for
    if min ≠ i then
        swap(A[i], A[min])           O(1)
end_for
```

$$f(n) = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} 1$$

(A) $$\sum_{i=j+1}^{n} 1 = n-(j+1)+1 = n-j \cancel{-1+1}$$
$$= n-j$$

(B) $$\sum_{j=1}^{n-1} n-j = n \cdot \sum_{j=1}^{n-1} 1 - \sum_{j=1}^{n-1} j$$

(B') $$n \cdot \sum_{j=1}^{n-1} = n(n-1 \cancel{-1+1}) = n(n-1) = n^2-n$$

(B'') $$\sum_{j=1}^{n-1} j = \frac{n-1(n-1+1)}{2} = \frac{n^2-n \cancel{-n+1} \cancel{+n-1}}{2}$$
$$= \frac{n^2-n}{2}$$

(B) $$n^2-n-\left(\frac{n^2-n}{2}\right) = \frac{2n^2-2n-n^2+n}{2}$$
(B) $$= \frac{n^2-n}{2}$$

Temos então que ...

$$f(n) = \frac{n^2-n}{2} \quad \therefore \quad O(n^2)$$

↳ função de custo ↳ complexidade