

O melhor caso é quando o array já está ordenado e o pior caso é quando o array está ordenado ao contrário. Porém o tempo de execução muda muito pouco, visto que apenas uma linha do código abaixo que deixa de ser executada quando a condição if não for satisfeita.

Análise dá primeira execução do for:

```
void selecao_direta(int numeros[], int N)
{
    int prox, iMenor, i;

    for (prox=0; prox < N - 1; prox++) // percorre vetor principal
    {
        iMenor = prox;

        for (i = prox + 1; i < N; i++)
        {
            if (numeros[i] < numeros[iMenor]) // se if não for satisfeita, não executa a linha a baixo e
                                                o tempo de execução diminui muito pouco

                iMenor = i; // quando este "for" terminar, iMenor terá o valor do índice do menor
                             elemento do vetor.
        }

        int temp = numeros[prox]; // temp grava o numero do primeiro elemento.

        numeros[prox] = numeros[iMenor]; //primeiro elemento passa a ter o menor valor que o
        restante à sua direita.

        numeros[iMenor] = temp; // o numero que tava na primeira posição é "colocado" onde
        tava o menor elemento.
    }
}
```

Os comentários foram feitos analisando apenas a primeira execução do for, ou seja quando prox=0, para quando prox =1,2,3... basta seguir o mesmo raciocínio, mudando a palavra "primeiro(a)" por "segundo(a)", "terceiro(a)" e assim por diante.