```
algoritmo uniaoOrdenada(v1, v2, n1, n2) {
{- Entrada: v1 e v2 são vetores de inteiros ordenados crescentemente, n1 e n2
são seus respectivos tamanhos
Saída: vetor união dos vetores v1 e v2 -}
        i := 1
        j := 1
        u := 1

        enquanto i < n1 && j < n2 faça
                se v1[i] < v2[j] então
                        uniao[u] = v1[i]
                        u := u+1
                        i := i+1
                senão se v1[i] > v2[j] então
                        uniao[u] = v2[j]
                        u := u+1
                        j := j+1
                senão então
                        uniao[u] = v1[i]
                        u := u+1
                        i := i+1
                        j := j+1

        enquanto i < n1 faça
                uniao[u] := v1[i]
                u := u+1
                i := i+1

        enquanto j < n2 faça
                uniao[u] := v2[j]
                u := u+1
                j := j+1

        retorne uniao
}
```

O espaço total utilizado da memória é:

- 1 inteiro para armazenar a variável i;
- 1 inteiro para armazenar a variável j;
- 1 inteiro para armazenar a variável u;
- 1 inteiro para armazenar a variável $n1$;

- 1 inteiro para armazenar a variável $n2$;
- $n1$ inteiros para armazenar os elementos de um vetor;
- $n2$ inteiros para armazenar os elementos do outro vetor.
  Complexidade de espaço:
  A quantidade total de memória usada pelo algoritmo uniaoOrdenada é $n1 + n2 + 5$, em função do tamanho da entrada.

Complexidade de tempo:
Considerando que $n1$ e $n2$ possuam o mesmo tamanho, para o pior caso a complexidade de tempo é dada por
$T(n) = 3T_{:=} + (T_{<} + T_{\&\&} + T_{<} + T_{<=}) + (T_{<} + T_{<})$
$= 3.1 + (n1 + n2 + 1)(1 + 1 + 1 + 1) + (1 + 1)$
$= 3.1 + (n1 + n2 + 1)(1 + 1 + 1 + 1) + (1 + 1)$
$= 3 + 4n1 + 4n2 + 4 + 2$
$= 4n1 + 4n2 + 9$
Considerando n como a soma das entradas $n1$ e $n2$ temos que a função que descreve a quantidade total de memória usada pelo algoritmo de união ordenada é $T(n) = 4n + 9$, que é linear em função do tamanho da entrada, deste modo, $T(n) = O(n)$.