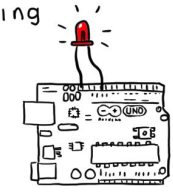


Just finished my first Arduino project:
A blinking led.



Next step:
Update my LinkedIn profile.

 Add Skill

Mechatronic
Engineer

turnoff.us

Merge

Paulo Ricardo Lisboa de Almeida

Faça você mesmo

função merge (v, a, m, b)

entrada: vetor v , indexado por $[a..b]$, onde $v[a..m]$ e $v[m+1..b]$ são vetores ordenados

saída: o vetor v modificado de forma que $v[a..b]$ é um vetor ordenado.

40 minutos

Faça o algoritmo iterativo em português ou C. O que você preferir.

Dica: Você pode criar um vetor auxiliar de $n = b - a + 1$ posições. Mas a resposta deve retornar no próprio vetor v .

Merge

função merge (v, a, m, b)

entrada: vetor v, indexado por [a..b], onde v[a..m] e v[m+1,b] são vetores ordenados

saída: o vetor v modificado de forma que v[a..b] é um vetor ordenado.

se $a \geq b$

 retorne

k \leftarrow 0

i \leftarrow a

j \leftarrow m + 1

enquanto k \leq b - a

 se j > b ou (i \leq m e v[i] \leq v[j])

 p \leftarrow i

 i \leftarrow i + 1

 senão

 p \leftarrow j

 j \leftarrow j + 1

 u[k] \leftarrow v[p]

 k \leftarrow k + 1

retorne copiar(v, a, b, u)

Merge

função merge (v,a,m,b)

entrada: vetor v, indexado por [a..b], onde v[a..m] e v[m+1,b] são vetores ordenados

saída: o vetor v modificado de forma que v[a..b] é um vetor ordenado.

se $a \geq b$

 retorne

k \leftarrow 0

i \leftarrow a

j \leftarrow m + 1

enquanto $k \leq b - a$

 se $j > b$ ou ($i \leq m$ e $v[i] \leq v[j]$)

 p \leftarrow i

 i \leftarrow i + 1

 senão

 p \leftarrow j

 j \leftarrow j + 1

 u[k] \leftarrow v[p]

 k \leftarrow k + 1

retorne copiar(v,a,b,u)

função copiar (v,u,a,b)

entrada: vetor v, indexado por [a..b], e vetor u indexado por [0,b-a]

saída: o vetor v modificado de forma que $v[a+i] = u[i] \forall i \in [0..b-a]$

i \leftarrow 0

enquanto $i \leq b - a$

 v[a+i] \leftarrow u[i]

 i \leftarrow i + 1

retorne

Teste de Mesa

i	1	2	3	4	5	6	7	8
v[i]	1	3	7	8	2	3	9	10

função merge (v, a, m, b)

entrada: vetor v, indexado por [a..b], onde v[a..m] e v[m+1,b] são vetores ordenados

saída: o vetor v modificado de forma que v[a..b] é um vetor ordenado.

se $a \geq b$

retorne

$k \leftarrow 0$

$i \leftarrow a$

$j \leftarrow m + 1$

enquanto $k \leq b - a$

se $j > b$ ou ($i \leq m$ e $v[i] \leq v[j]$)

$p \leftarrow i$

$i \leftarrow i + 1$

senão

$p \leftarrow j$

$j \leftarrow j + 1$

$u[k] \leftarrow v[p]$

$k \leftarrow k + 1$

retorne copiar(v, a, b, u)

função copiar (v, u, a, b)

entrada: vetor v, indexado por [a..b], e vetor u indexado por [0, b-a]

saída: o vetor v modificado de forma que $v[a+i] = u[i] \forall i \in [0..b-a]$

$i \leftarrow 0$

enquanto $i \leq b - a$

$v[a+i] \leftarrow u[i]$

$i \leftarrow i + 1$

retorne

Análise

Número de comparações entre elementos do vetor

Precisamos envolver a função copiar nessa análise?

função merge (v, a, m, b)

se $a \geq b$

retorne

$k \leftarrow 0$

$i \leftarrow a$

$j \leftarrow m + 1$

enquanto $k \leq b - a$

se $j > b$ ou ($i \leq m$ e $v[i] \leq v[j]$)

$p \leftarrow i$

$i \leftarrow i + 1$

senão

$p \leftarrow j$

$j \leftarrow j + 1$

$u[k] \leftarrow v[p]$

$k \leftarrow k + 1$

retorne copiar(v, a, b, u)

função copiar (v, u, a, b)

$i \leftarrow 0$

enquanto $i \leq b - a$

$v[a+i] \leftarrow u[i]$

$i \leftarrow i + 1$

retorne v

Análise

Número de comparações entre elementos do vetor

Precisamos envolver a função copiar nessa análise?

Não. A função copiar não faz comparações.

função merge (v, a, m, b)

se $a \geq b$

retorne

$k \leftarrow 0$

$i \leftarrow a$

$j \leftarrow m + 1$

enquanto $k \leq b - a$

se $j > b$ ou ($i \leq m$ e $v[i] \leq v[j]$)

$p \leftarrow i$

$i \leftarrow i + 1$

senão

$p \leftarrow j$

$j \leftarrow j + 1$

$u[k] \leftarrow v[p]$

$k \leftarrow k + 1$

retorne copiar(v, a, b, u)

função copiar (v, u, a, b)

$i \leftarrow 0$

enquanto $i \leq b - a$

$v[a+i] \leftarrow u[i]$

$i \leftarrow i + 1$

retorne v

Análise

Número de comparações entre elementos do vetor

Quantas vezes passamos pelo loop?

```
função merge (v, a, m, b)  
se a ≥ b  
    retorne v  
k ← 0  
i ← a  
j ← m + 1  
enquanto k ≤ b - a  
    se j > b ou (i ≤ m e v[i] ≤ v[j])  
        p ← i  
        i ← i + 1  
    senão  
        p ← j  
        j ← j + 1  
    u[k] ← v[p]  
    k ← k + 1  
retorne copiar(v, a, b, u)
```


Análise

Número de comparações entre elementos do vetor

Quantas vezes passamos pelo loop?

Lembre-se que $n = b - a + 1$

Logo o loop é executado enquanto $k \leq n - 1$

Como k inicia em 0, o loop é executado **n vezes**

```
função merge (v, a, m, b)
se a ≥ b
    retorne v
k ← 0
i ← a
j ← m + 1
enquanto k ≤ b - a
    se j > b ou (i ≤ m e v[i] ≤ v[j])
        p ← i
        i ← i + 1
    senão
        p ← j
        j ← j + 1
    u[k] ← v[p]
    k ← k + 1
retorne copiar(v, a, b, u)
```

Análise

Número de comparações entre elementos do vetor

Quantas comparações entre elementos do vetor são feitas a cada iteração?

```
função merge (v, a, m, b)
  se a ≥ b
    retorne v
  k ← 0
  i ← a
  j ← m + 1
  enquanto k ≤ b - a
    se j > b ou (i ≤ m e v[i] ≤ v[j])
      p ← i
      i ← i + 1
    senão
      p ← j
      j ← j + 1
    u[k] ← v[p]
    k ← k + 1
  retorne copiar(v, a, b, u)
```

Análise

Número de comparações entre elementos do vetor

Quantas comparações entre elementos do vetor são feitas a cada iteração?

Uma comparação

Atenção: aqui vamos fazer uma simplificação

A comparação é sempre feita (não há curto circuito)

```
função merge (v, a, m, b)
se a ≥ b
    retorne v
k ← 0
i ← a
j ← m + 1
enquanto k ≤ b - a
    se j > b ou (i ≤ m e v[i] ≤ v[j])
        p ← i
        i ← i + 1
    senão
        p ← j
        j ← j + 1
    u[k] ← v[p]
    k ← k + 1
retorne copiar(v, a, b, u)
```

Análise

Número de comparações entre elementos do vetor

Quantas comparações entre elementos do vetor são feitas a cada iteração?

Uma comparação

Atenção: aqui vamos fazer uma simplificação

A comparação é sempre feita (não há curto circuito)

Logo

$$C_m(n) = n$$

```
função merge (v, a, m, b)
se a ≥ b
    retorne v
k ← 0
i ← a
j ← m + 1
enquanto k ≤ b - a
    se j > b ou (i ≤ m e v[i] ≤ v[j])
        p ← i
        i ← i + 1
    senão
        p ← j
        j ← j + 1
    u[k] ← v[p]
    k ← k + 1
retorne copiar(v, a, b, u)
```

Custo de memória

Considerando $n = b - a + 1$ o tamanho do vetor v , quanta memória é consumida para armazenar v e vetores auxiliares?

```
função merge (v, a, m, b)
se a ≥ b
    retorne v
k ← 0
i ← a
j ← m + 1
enquanto k ≤ b - a
    se j > b ou (i ≤ m e v[i] ≤ v[j])
        p ← i
        i ← i + 1
    senão
        p ← j
        j ← j + 1
    u[k] ← v[p]
    k ← k + 1
retorne copiar(v, a, b, u)
```

Custo de memória

Considerando $n = b - a + 1$ o tamanho do vetor v , quanta memória é consumida para armazenar v e vetores auxiliares?

$$M_m(n) = \begin{cases} 0, & \text{se } n \leq 1, \\ 2n, & \text{se } n > 1 \end{cases}$$

Vetor auxiliar u de n posições

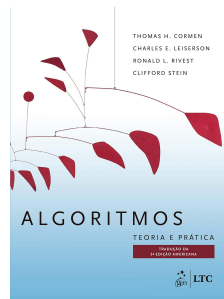
```
função merge (v, a, m, b)
se a ≥ b
  retorne v
k ← 0
i ← a
j ← m + 1
enquanto k ≤ b - a
  se j > b ou (i ≤ m e v[i] ≤ v[j])
    p ← i
    i ← i + 1
  senão
    p ← j
    j ← j + 1
  u[k] ← v[p]
  k ← k + 1
retorne copiar(v, a, b, u)
```

Exercícios

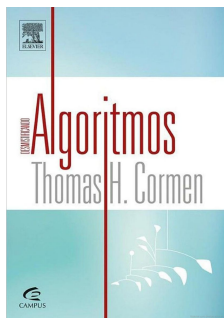
1. Implemente o algoritmo de merge discutido em aula em C.

Referências

T. Cormen, C. Leiserson,
R. Rivest, C. Stein.
Algoritmos: Teoria e
Prática. 3a ed. 2012



T. Cormen.
Desmistificando
algoritmos. 2017.

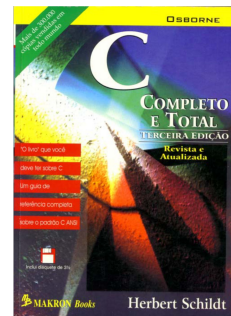


Renato Carmo. Algoritmos e
Estruturas de Dados.
www.inf.ufpr.br/renato

R. Sedgwick, K. Wayne.
Algorithms Part I. 4a ed.
2014



H. Schildt. C completo e
total. 1996



Licença

Este obra está licenciado com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

