

Estrutura de Dados Avançada

Daniel de Sousa Moraes
danielmoraes14@gmail.com

Merge Sort

- Divisão e Conquista: Dividir, Conquistar, Combinar
- Divide a sequência de **N** elementos em subsequências menores
- Ordena as subsequências menores utilizando ordenação por intercalação
- Combina as subsequências para produzir a resposta ordenada

Merge Sort

- Ordena inicialmente vetores de tamanho 1 ou 2 e recursivamente dobra esse tamanho até ordenar o vetor total de tamanho N
- Baseia-se no merge de dois vetores já ordenados

Merge de dois vetores

- Dados os vetores $V1$ e $V2$ já ordenados, o merge cria um novo vetor V ordenado com os elementos de $V1$ e $V2$

Merge de dois vectores

```
Merge(V1,V2)
  i,j,k = 0
  for k = 0, k < #V1+#V2 do
    if i >= #V1 then V[k]=V2[ j ]; j++;
    elseif j >= #V2 then V[k]=V1[ i ]; i++;
    elseif V1[ i ] <= V2[ j ] then V[k]=V1[i]; i++;
    else V[k] = V2[j]; j++;
  end
end
end
```

Merge

$V1=$

i					
	3	5	7	8	10

$V2=$

j					
	1	2	4	6	9

$V=$

k									
	1								

Merge

$V1=$

i					
	3	5	7	8	10

$V2=$

	j				
	1	2	4	6	9

$V=$

	k								
	1	2							

Merge

V1=

i	3	5	7	8	10
---	---	---	---	---	----

V2=

j	1	2	4	6	9
---	---	---	---	---	---

V=

k	1	2	3						
---	---	---	---	--	--	--	--	--	--

Merge

V1=

		i		
3	5	7	8	10

V2=

		j		
1	2	4	6	9

V=

			k						
1	2	3	4						

Merge

V1=

		i		
3	5	7	8	10

V2=

			j	
1	2	4	6	9

V=

				k					
1	2	3	4	5					

Merge

V1=

3	5	7	8	10
---	---	---	---	----

i

V2=

1	2	4	6	9
---	---	---	---	---

j

V=

1	2	3	4	5	6				
---	---	---	---	---	---	--	--	--	--

k

Merge

V1=

3	5	ⁱ 7	8	10
---	---	-------------------	---	----

V2=

1	2	4	6	^j 9
---	---	---	---	-------------------

V=

1	2	3	4	5	6	^k 7			
---	---	---	---	---	---	-------------------	--	--	--

Merge

V1=

3	5	7	ⁱ 8	10
---	---	---	-------------------	----

V2=

1	2	4	6	^j 9
---	---	---	---	-------------------

V=

1	2	3	4	5	6	7	^k 8		
---	---	---	---	---	---	---	-------------------	--	--

Merge

V1=

3	5	7	8	10
---	---	---	---	----

ⁱ

V2=

1	2	4	6	9
---	---	---	---	---

^j

V=

1	2	3	4	5	6	7	8	9	
---	---	---	---	---	---	---	---	---	--

^k

Merge

V1=

3	5	7	8	10
---	---	---	---	----

i

V2=

1	2	4	6	9
---	---	---	---	---

j

V=

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

k

Merge

V1=

3	5	7	8	10
---	---	---	---	----

i

V2=

1	2	4	6	9
---	---	---	---	---

j

V=

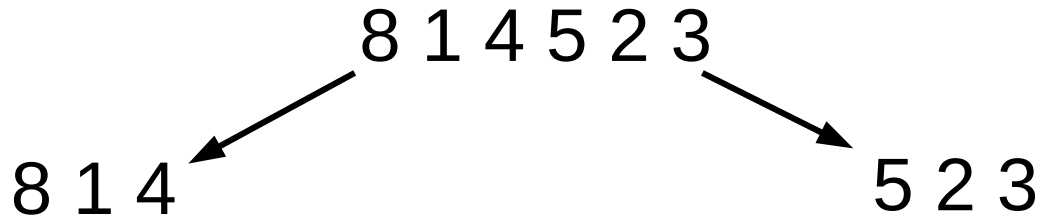
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

k

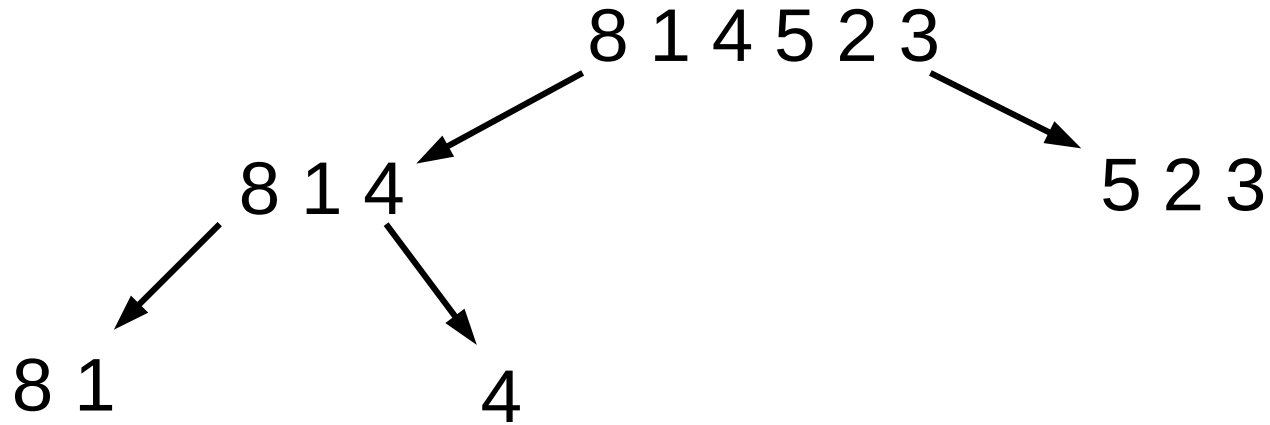
Merge Sort

8 1 4 5 2 3

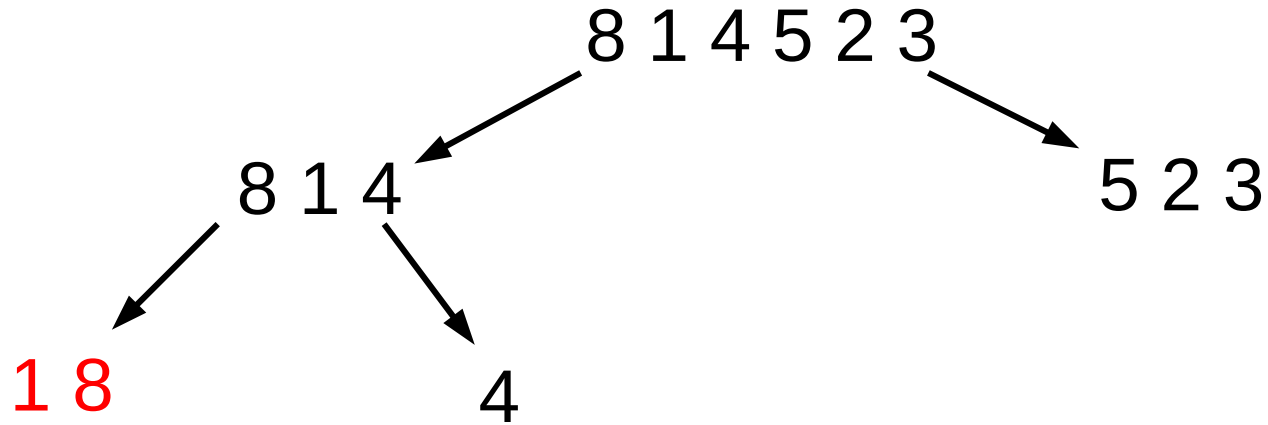
Merge Sort



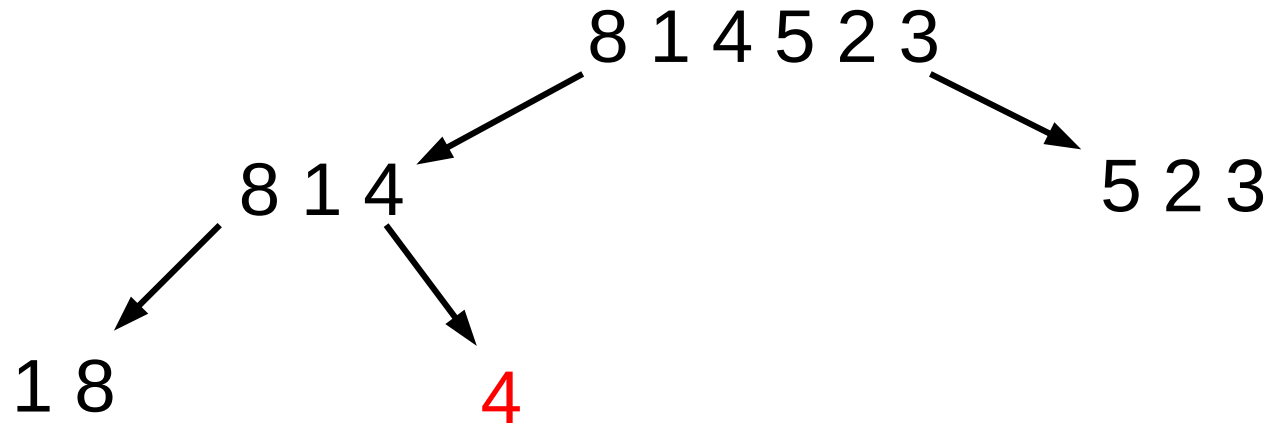
Merge Sort



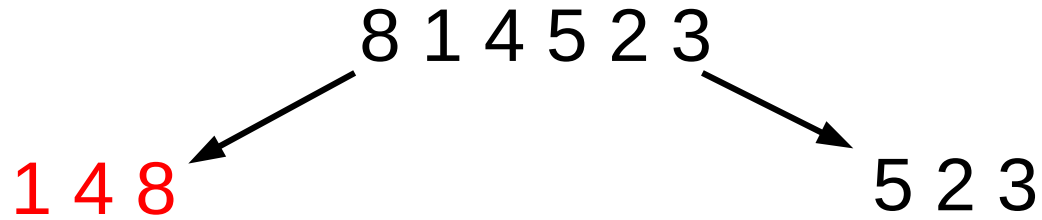
Merge Sort



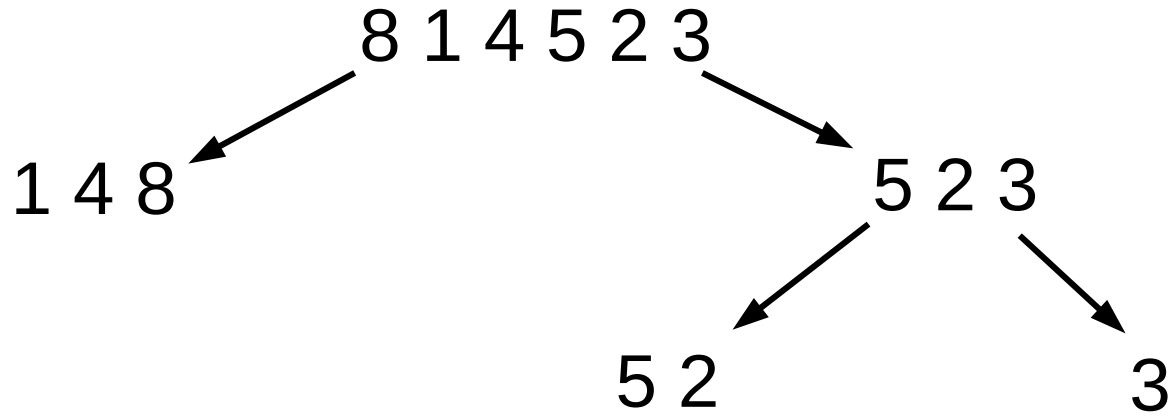
Merge Sort



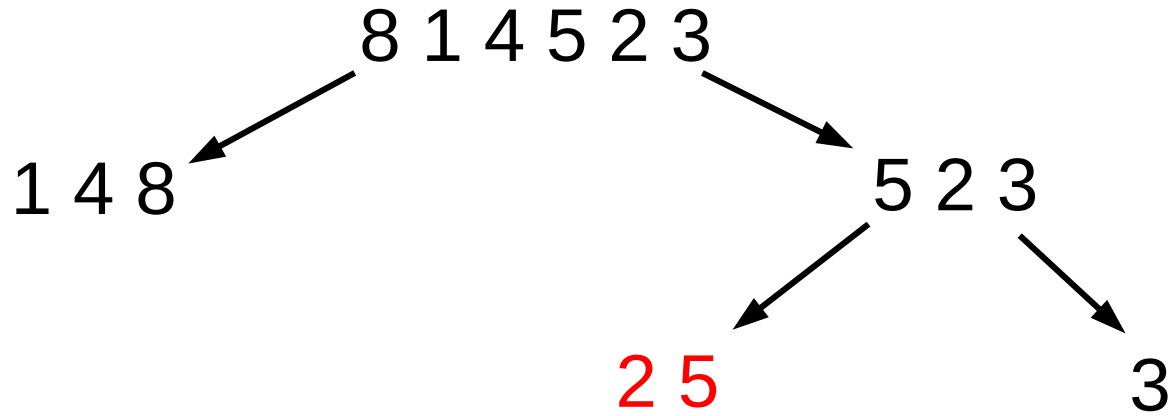
Merge Sort



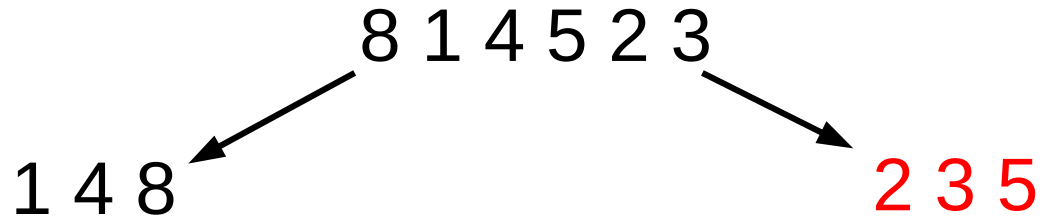
Merge Sort



Merge Sort



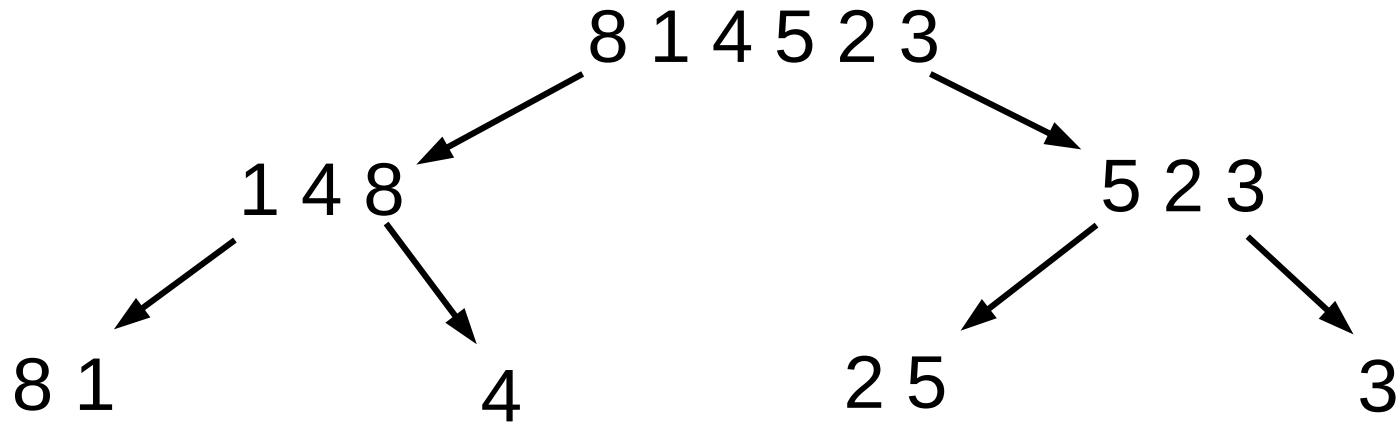
Merge Sort



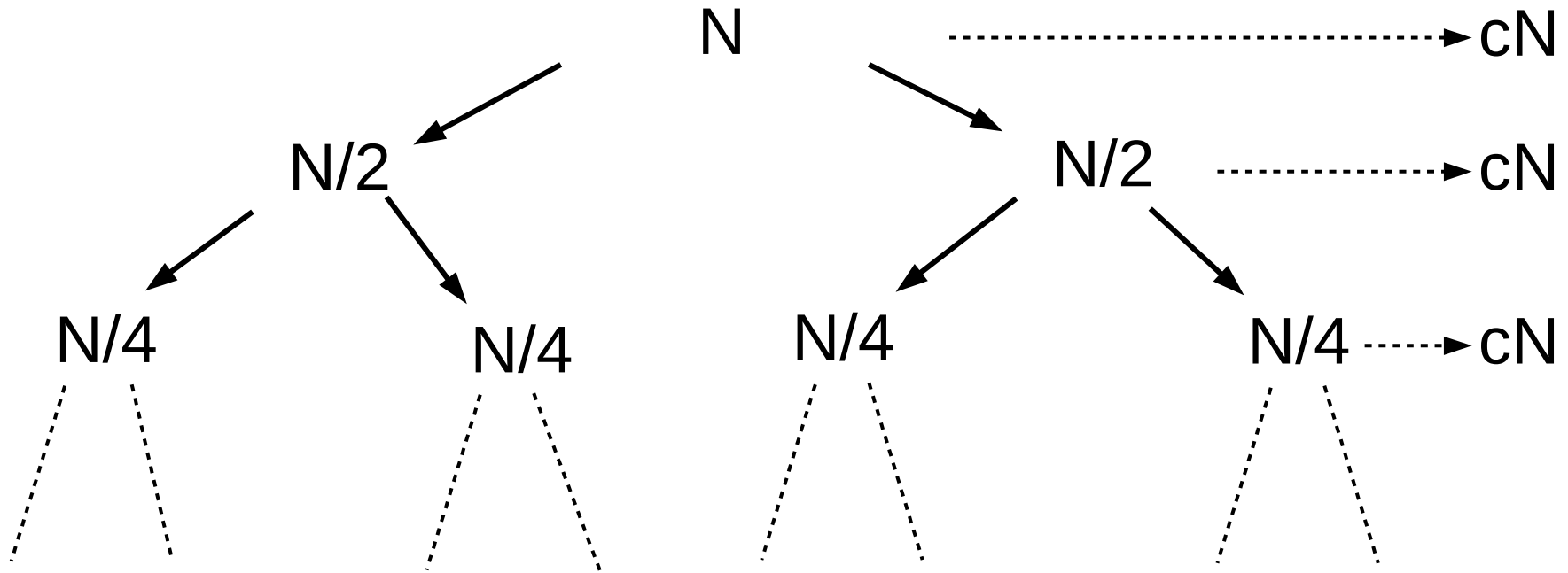
Merge Sort

1 2 3 4 5 8

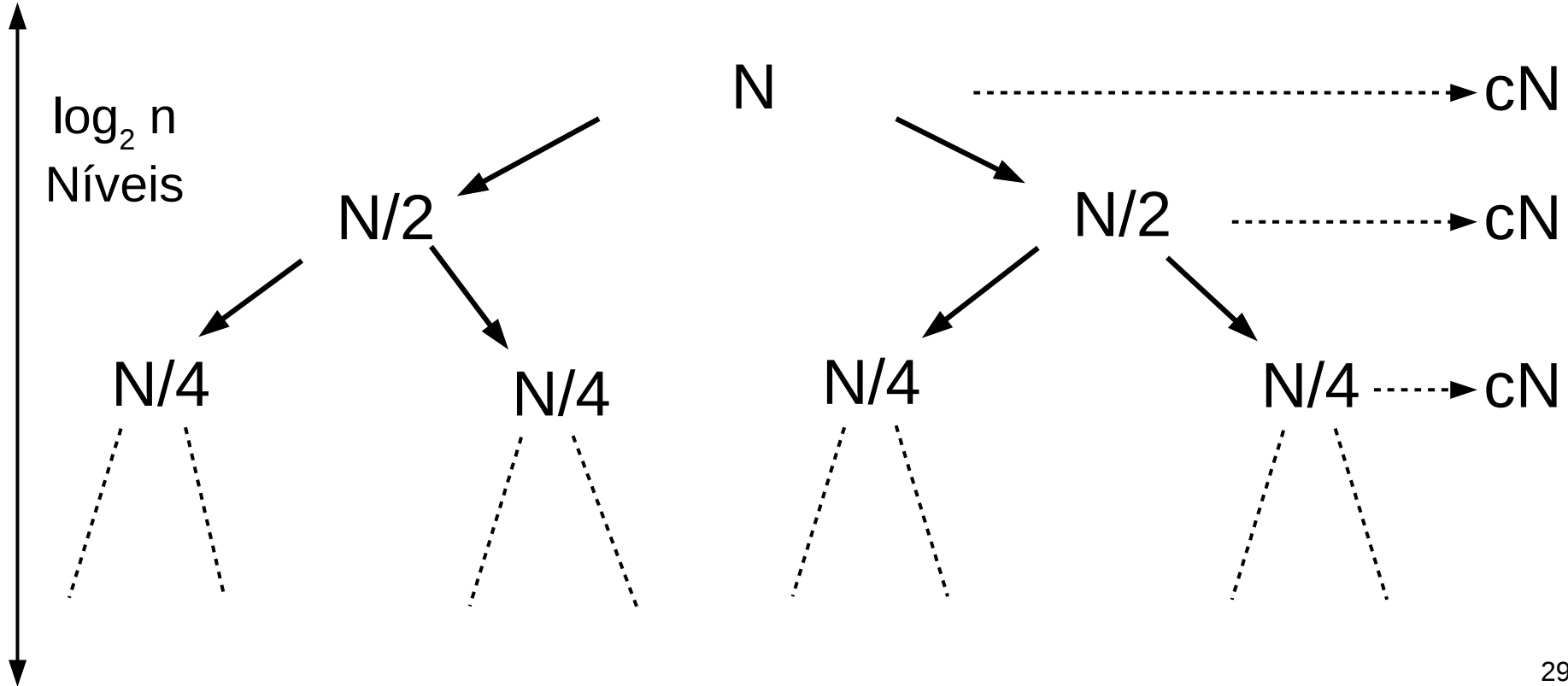
Merge Sort - operações



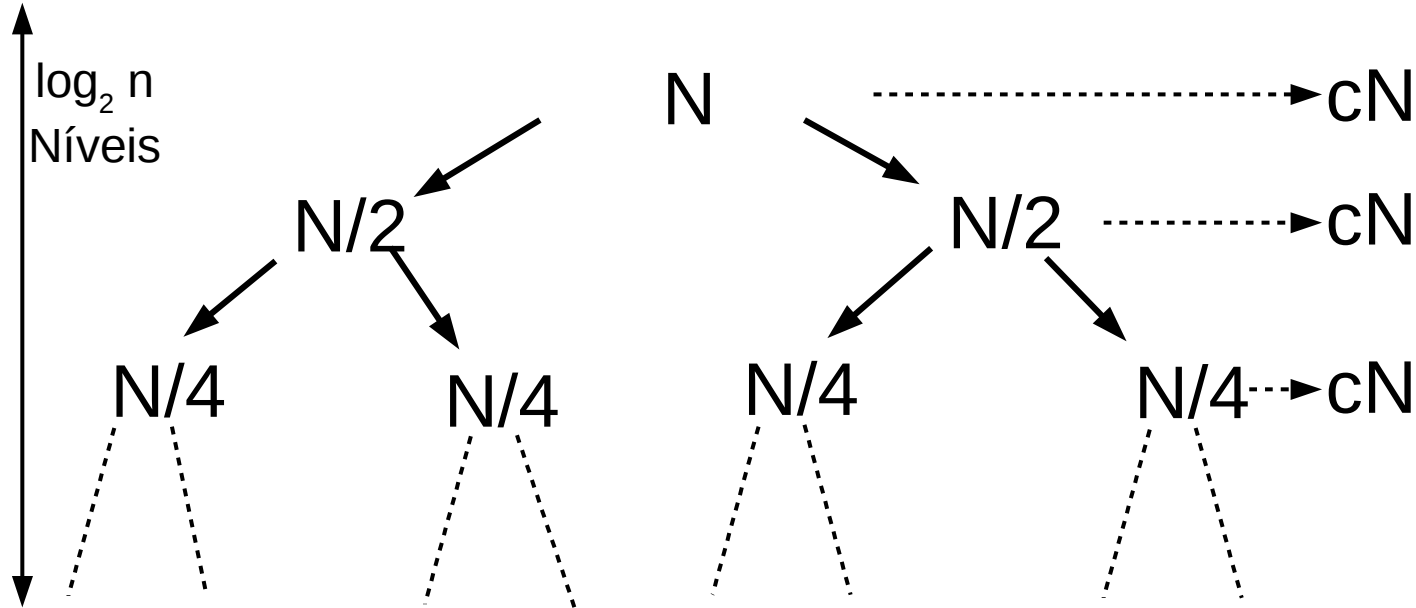
Merge Sort - operações



Merge Sort - operações



Merge Sort - operações



$$N^{\circ} \text{ de operações} = N * \log N$$

Bibliografia

Cormen, Thomas H. et al. Algoritmos.; [tradução Arlete Simille]. 3ª ed - Rio de Janeiro - Elsevier, 2011.

Soares Neto, Carlos de Salles. Notas de Aula
Disciplina EDII - UFMA