



ALGORITMOS E COMPLEXIDADE (ARA0174)

AULA 8

DATA: 07/04/2022



Aula 8- ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMO DE ORDENAÇÃO AVANÇADOS (MEGERSORT)

CONTINUAÇÃO...



Estácio

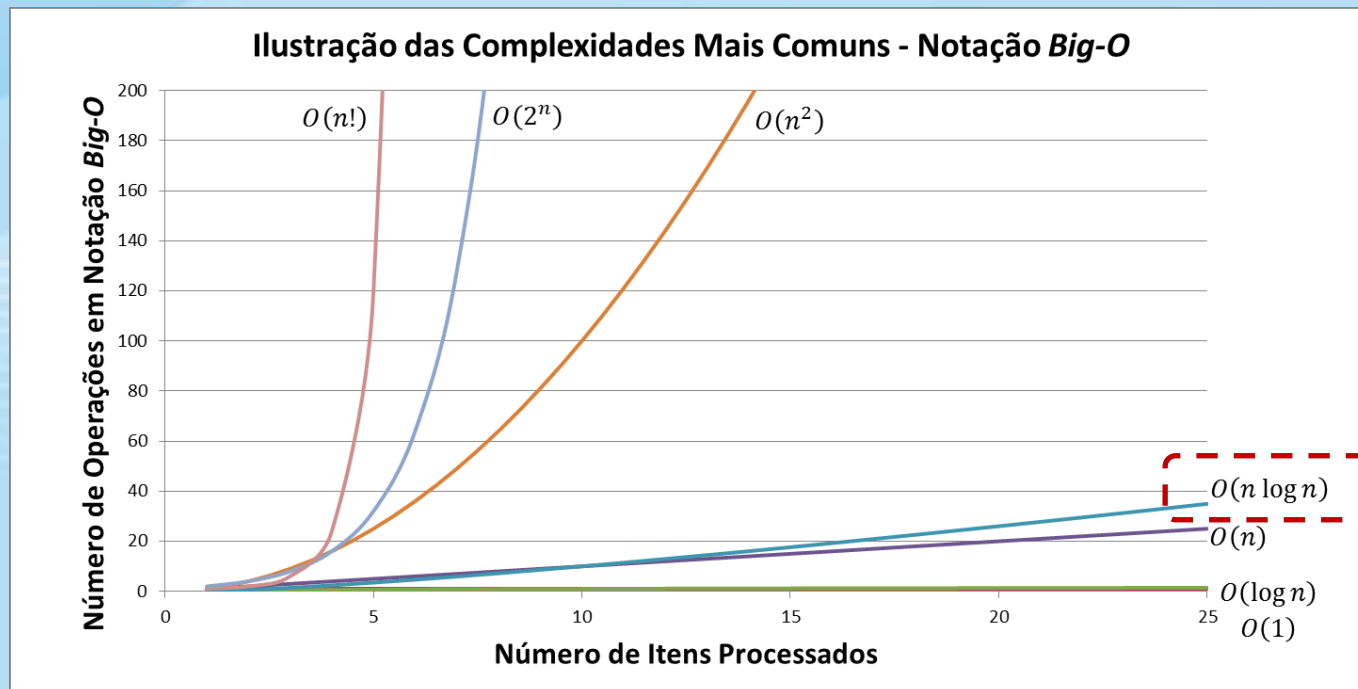
Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

ALGORITMO DE ORDENAÇÃO POR INTERCALAÇÃO (MERGESORT)

Independentemente se os elementos estão ordenados OU não, a complexidade é sempre a mesma.

- *Melhor caso: $O(n \log n)$*
- *Pior caso: $O(n \log n)$*



Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

ALGORITMO DE ORDENAÇÃO POR INTERCALAÇÃO (MERGESORT)

Características do MERGESORT (Análise)

- No *MERGESORT* dividimos recursivamente pela metade os elementos, estabelecendo uma *árvore de decisão*.
- A *arvore de decisão* representa as possíveis comparações que foram realizadas no algoritmos de ordenação.
- Depois fomos recombinação os dados ordenados até obtermos um *conjunto só*.

Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

ALGORITMO DE ORDENAÇÃO POR INTERCALAÇÃO (MERGESORT)

Antes de trabalharmos com o algoritmo de ordenação por intercalação MERGESORT, vamos reforçar o entendimento de **RECURSIVIDADE**, pois o algoritmo em questão utiliza.

Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

RELEMBRANDO RECURSIVIDADE: EXERCÍCIO 1

```
1  #include<iostream>
2  #include<locale.h>
3  using namespace std;
4
5  int dec2bin(int dec);
6
7  int main() {
8      setlocale(LC_ALL, "portuguese");
9      int dec;
10     cout << "Digite um no. inteiro: ";
11     cin >> dec;
12     cout << "\n\n(Leia da direita para a esquerda)";
13     cout << "\n Resultado: ";
14     dec2bin(dec);
15     return 0;
16 }
17
18 int dec2bin(int dec) {
19     int n, resto;
20     if (dec/2!=0) {
21         n = dec/2;
22         resto = dec%2;
23         cout << resto << " ";
24         return dec2bin(n);
25     } else {
26         resto = dec%2;
27         cout << resto << " ";
28     }
29 }
```

Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

RELEMBRANDO RECURSIVIDADE: EXERCÍCIO 1

```
1  #include<iostream>
2  #include<locale.h>
3  using namespace std;
4
5  int dec2bin(int dec);
6
7  int main() {
8      setlocale(LC_ALL, "portuguese");
9      int dec;
10     cout << "Digite um no. inteiro: ";
11     cin >> dec;
12     cout << "\n\n(Leia da direita para a esquerda)";
13     cout << "\n Resultado: ";
14     dec2bin(dec);
15     return 0;
16 }
17
18 int dec2bin(int dec) {
19     int n, resto;
20     if (dec/2!=0) {
21         n = dec/2;
22         resto = dec%2;
23         cout << resto << " ";
24         return dec2bin(n);
25     } else {
26         resto = dec%2;
27         cout << resto << " ";
28     }
29 }
```

Mas como fazer exibir na ordem correta?
O que devo utilizar?

Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

RELEMBRANDO

RECURSIVIDADE: EXERCÍCIO 1

Deste modo consigo exibir
na ordem.

```
1  #include<iostream>
2  #include<locale.h>
3  using namespace std;
4
5  int dec2bin(int dec);
6
7  int main() {
8      setlocale(LC_ALL, "portuguese");
9      int dec;
10     cout << "Digite um no. inteiro: ";
11     cin >> dec;
12     cout << "\n\n(Leia da direita para a esquerda)";
13     cout << "\n Resultado: ";
14     dec2bin(dec);
15     return 0;
16 }
17
18 int dec2bin(int dec) {
19     int n, resto;
20     static int vetor[10], i=0, k;
21
22     if (dec/2!=0) {
23         n = dec/2;
24         resto = dec%2;
25         vetor[i]=resto;
26         cout << resto << " ";
27         i=i+1;
28         return dec2bin(n);
29     } else {
30         resto = dec%2;
31         vetor[i]=resto;
32         cout << resto << " ";
33     }
34     cout << "\n\n Exibindo na ordem: ";
35     for (k=i; k>=0; k--) {
36         cout << vetor[k] << " ";
37     }
38 }
```


Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

RELEMBRANDO

RECURSIVIDADE: EXERCÍCIO 1

E se o objetivo agora for de decimal para OCTAL? O que eu mudaria no código?

```
1  #include<iostream>
2  #include<locale.h>
3  using namespace std;
4
5  int dec2bin(int dec);
6
7  int main() {
8      setlocale(LC_ALL, "portuguese");
9      int dec;
10     cout << "Digite um no. inteiro: ";
11     cin >> dec;
12     cout << "\n\n(Leia da direita para a esquerda)";
13     cout << "\n Resultado: ";
14     dec2bin(dec);
15     return 0;
16 }
17
18 int dec2bin(int dec) {
19     int n, resto;
20     static int vetor[10], i=0, k;
21
22     if (dec/2!=0) {
23         n = dec/2;
24         resto = dec%2;
25         vetor[i]=resto;
26         cout << resto << " ";
27         i=i+1;
28         return dec2bin(n);
29     } else {
30         resto = dec%2;
31         vetor[i]=resto;
32         cout << resto << " ";
33     }
34     cout << "\n\n Exibindo na ordem: ";
35     for (k=i; k>=0; k--) {
36         cout << vetor[k] << " ";
37     }
38 }
```

Aula 8 - ALGORITMOS E COMPLEXIDADE

TEMA 3 - ALGORITMOS DE ORDENAÇÃO AVANÇADOS

RELEMBRANDO

RECURSIVIDADE: EXERCÍCIO 1

Agora o código está trabalhando na conversão de decimal para OCTAL.

```
1  #include<iostream>
2  #include<locale.h>
3  using namespace std;
4
5  int dec2oct(int dec);
6
7  int main() {
8      setlocale(LC_ALL, "portuguese");
9      int dec;
10     cout << "Digite um no. inteiro: ";
11     cin >> dec;
12     cout << "\n\n(Leia da direita para a esquerda)";
13     cout << "\n Resultado: ";
14     dec2oct(dec);
15     return 0;
16 }
17
18 int dec2oct(int dec) {
19     int n, resto;
20     static int vetor[10], i=0, k;
21
22     if (dec/8!=0) {
23         n = dec/8;
24         resto = dec%8;
25         vetor[i]=resto;
26         cout << resto << " ";
27         i=i+1;
28         return dec2oct(n);
29     } else {
30         resto = dec%8;
31         vetor[i]=resto;
32         cout << resto << " ";
33     }
34     cout << "\n\n Exibindo na ordem: ";
35     for (k=i; k>=0; k--) {
36         cout << vetor[k] << " ";
37     }
38 }
```