

# Comparando os algoritmos de ordenação

Ruben Carlo Benante

Autor2

Autor3

22 de Abril de 2021

## Resumo

Vamos comparar os algoritmos *xsort* e *ysort* para bla bla. Para compilar o PDF a partir do TEX use o comando:

1. `pdflatex exN.tex -o exN.pdf`
2. `bibtex exN`
3. `pdflatex exN.tex -o exN.pdf`

Este é o fim do resumo.

## 1 Introdução ao métodos de ordenação

Métodos de ordenação tem a principal função de...

O algoritmo *Tal-e-qual* trabalha percorrendo uma árvore tal tal e tal.

## 2 O *BubbleSort*

O algoritmo *Xsort* trabalha fazendo uma varredura na ...

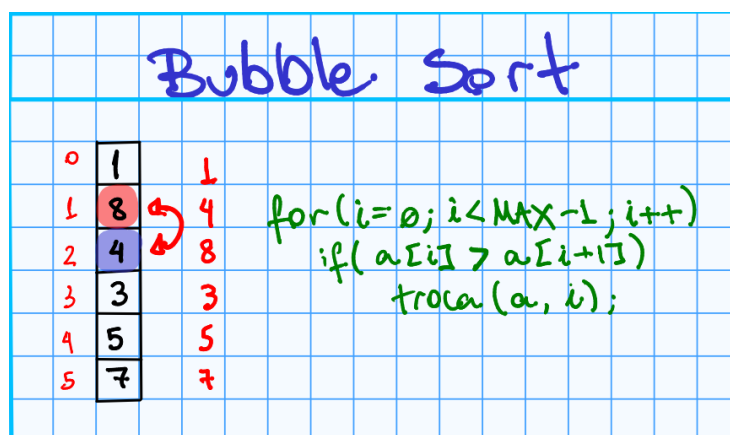


Figura 1: Exemplo de ordenação com Bubblesort

## 2.1 Implementação

O algoritmo é descrito abaixo:

## 3 O *QuickSort*

O método *Ysort* é caracterizado por...

### 3.1 Implementação

Para conseguir blablabla

O algoritmo *Ysort* segue abaixo:

---

**Algoritmo 1** Algoritmo Ysort

---

```
1: function YSORT(estado)                                ▷ retorna uma ação
2:   Entradas: estado é a configuração atual do jogo
3:    $v \leftarrow \text{maxvalor}(\text{estado})$ 
4:   retorna a ação  $a$  em sucessores(estado) cujo valor é  $v$ 
5: end function
6: function MAXVALOR(estado)                              ▷ retorna o valor estático
7:   if fim(estado) then
8:     retorna estatico(estado)
9:   end if
10:   $v \leftarrow -\infty$ 
11:  for todas ações  $a$  nos sucessores(estado) do
12:     $v \leftarrow \max(v, \text{minvalor}(a))$ 
13:  end for
14:  retorna  $v$ 
15: end function
16: function MINVALOR(estado)                              ▷ retorna o valor estático
17:   if fim(estado) then
18:     retorna estatico(estado)
19:   end if
20:   $v \leftarrow \infty$ 
21:  for todas ações  $a$  nos sucessores(estado) do
22:     $v \leftarrow \min(v, \text{maxvalor}(a))$ 
23:  end for
24:  retorna  $v$ 
25: end function
```

---

## 4 Comparando *XSort* o *YSort*

### 4.1 Método

O método usado para comparação foi...

### 4.2 Resultados

Os resultados mostrados na tabela 1 demonstram ...

Tabela 1: Tabela de custo de pontos para habilidades

pontos	moedas
8	0
9	1
10	2
11	3
12	4
13	5
14	7
15	9

## 5 Conclusão do artigo

Concluimos, com base nos estudos e testes coletados sobre os algoritmos de ordenação propostos, que para fins educacionais, o algoritmo *BubbleSort* é mais indicado devido a sua simples implementação, cabendo então para o *QuickSort* ser o mais indicado entre os dois, quando requer uma demanda em menor tempo e com mais eficiência.

De acordo com [?], este é o fim do artigo.

## Referências

- [1] BENANTE, R. C. *Geração de Trajetórias de Estados por Mapas Auto-organizáveis com Topologia Dinâmica*. Doutorado em ciências da computação, Universidade Federal de Pernambuco, Recife, 2008.