

ProgramAuto: subtítulo

Ruben Carlo Benante

Autor2

Autor3

13 de Junho de 2021

Resumo

Assunto: Ensino da Linguagem de Programação C.

Vamos comparar os algoritmos *xsort* e *ysort* para bla bla.

Local: Escola Politécnica de Pernambuco - UPE/POLI

Órgão Financiador: N/A

Caracterização: Projeto de Extensão requisito da disciplina de Matemática Discreta, sub-projeto integrante do Projeto **ProgramAuto**

1 Introdução

Ensino da Linguagem de Programação C

Métodos de ordenação tem a principal função de...

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

2 Objetivos

2.1 Objetivo Geral

Descrever o objetivo geral a ser alcançado

2.2 Objetivos Específicos

Listar os objetivos específicos

- Proporcionar tal e tal
- Realizar tal e tal

3 Justificativa

Justificar seu projeto.

4 Metodologia

Descrever como (por quais métodos) os objetivos serão alcançados.

O algoritmo é descrito abaixo:

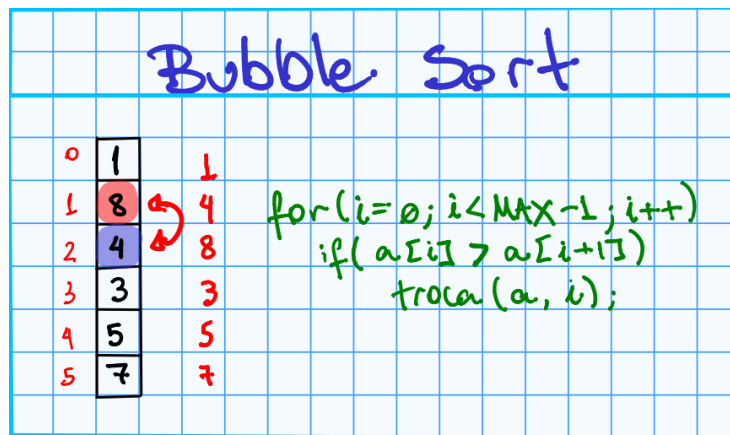


Figura 1: Exemplo de ordenação com Bubblesort

4.1 Equipamentos Necessários

Para realizar este projeto é preciso tal e tal
O método *Ysort* é caracterizado por...

4.2 Implementação

Para conseguir blablabla
O algoritmo *Ysort* segue abaixo:

5 Plano de Trabalho

Esta seção estabelece as atividades a serem realizadas.

6 Cronograma

Em conjunto com a seção de Plano de Trabalho, a seção de cronograma coloca as atividades dispostas numa linha do tempo.
Utilize uma tabela para melhor visualização.

7 Impactos e Transferências

7.1 Impacto Científico

Não há impacto científico relevante.

7.2 Impacto Tecnológico

Não há impacto tecnológico relevante.

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
```

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

7.3 Impacto Econômico

Não há impacto econômico relevante.

7.4 Impacto Social

O projeto visa contribuir com a sociedade na forma de... bla ... bla... blal

7.5 Impacto Ambiental

Não há impacto ambiental relevante.

7.6 Transferências

O ProgramAuto terá como objetivo a execução de aulas para ensinar a linguagem C aos discentes interessados. Contará com sua exposição de ensino gravada que será disponibilizada e a elaboração de relatórios a fim de cumprir com os aspectos estabelecidos no Plano de Trabalho.

O que mais o seu projeto agrega? O que é transferido? De onde vem? Para onde vai?

8 Resultados Esperados

Os resultados mostrados na tabela 1 demonstram ...

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 [1], 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.