Estrutura de Dados

Pollyana Roberta de Sousa

Questão

P.3.17. Construa um sistema que ordene valores em uma lista de objetos contendo os atributos nome:string e idade:número. Seu sistema deve ordenar os valores primeiramente pelo nome e, posteriormente, pela idade. Note que seu algoritmo de ordenação deverá operar em 2 camadas. A primeira ordena pelo nome e a segunda ordena os objetos de mesmo nome mas utilizando o telefone. Faça a devida adaptação no algoritmo de ordenação da bolha fornecido pelo professor.

P.3.18. Repita o problema 3.17 mas utilizando o algoritmo da inserção

Explicação do Algoritmo de Inserção para Ordenação Alfabética

O algoritmo de inserção funciona comparando um elemento com os anteriores da lista e deslocando-os até encontrar a posição correta para o elemento atual. Ele é eficiente para listas pequenas e funciona da seguinte forma:

1. **Começa do segundo elemento** da lista e o compara com o anterior.
2. **Desloca os elementos** maiores (neste caso, os que vêm depois em ordem alfabética) para a frente, até encontrar o lugar correto para o elemento atual.
3. **Insere o elemento na posição correta** e repete o processo até o final da lista.

No exemplo da lista de pessoas com nomes, estamos ordenando **alfabeticamente** os nomes. A cada vez que um nome é deslocado, o estado da lista é exibido para mostrar o que está acontecendo.

No final, todos os elementos estão em ordem alfabética crescente.

Vamos pro Código

```
interface Pessoa {
    nome: string;
    idade: number;
}
```

1) Define o formato dos objetos que serão manipulados. Cada pessoa tem dois atributos: **nome** e **idade.**

```
function insercao(v: Pessoa[]): Pessoa[] {
```

2) Essa função recebe uma array de objetos **Pessoa,** faz a ordenação e retorna uma array ordenada.

```
for (let i: number = 1; i < v.length; i++) {
  let elemento: Pessoa = v[i]; //O elemento que está sendo inserido na posição correta
  let pos: number = i;
```

3) O laço **for** começa na posição 1 do array (i = 1) e percorre até o final do array. A variável **elemento** representa o objeto Pessoa que está sendo inserido na posição correta dentro da parte já ordenada do array. A variável **pos** é usada para controlar a posição em que o elemento será inserido.

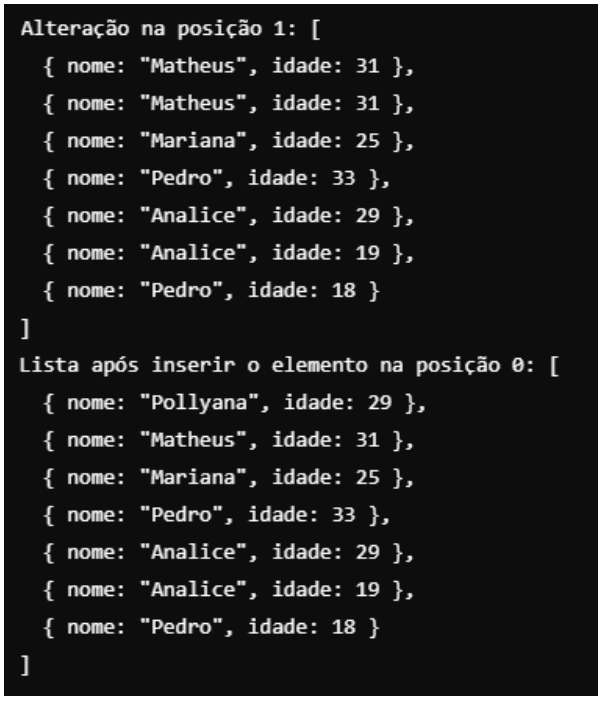
```
// Enquanto a posição for maior que 0 e o nome anterior for maior que o nome atual(em ordem alfabética e por idade)
while (pos > 0 && (v[pos - 1].nome > elemento.nome ||
        (v[pos - 1].nome === elemento.nome && v[pos - 1].idade > elemento.idade))) {
  v[pos] = v[pos - 1]; // Move o elemento anterior para a direita
  pos = pos - 1; // Reduz uma posição

  //(só pra fazer debug)
  //console.log(`Alteração na posição ${pos + 1}:`, v);
}
```

4) O laço **while** compara o elemento atual (**elemento**) com os elementos anteriores no array (**v [pos-1]**). Ele continua a mover os elementos para a direita enquanto o nome ou a idade for maior do que o nome ou idade do elemento.

```
v[pos] = elemento; //Insere o elemento na posição correta
```

5) Ao reduzir uma posição, o elemento pode ser inserido na posição correta.



6) A primeira ordenação é temporária (antes de inserir de fato o elemento na posição correta)

FIM =)