



UNIVERSIDADE DE SÃO PAULO - ICMC  
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO  
SCC0220 "Lab. ICC2" - 2ºSem/2020

## *Exercício 06 - Sort PokeDex*

### **Tarefa run codes:**

Usando os arquivos contidos no zip "Código" você deve criar um programa que leia um inteiro indicando o tipo de algoritmo de sort a ser usado, seguido de uma string indicando o atributo do pokemon a ser usado pelo algoritmo, seguido por uma terceira string representando o nome do arquivo que contém os dados dos pokemons. Você ordenará a PokeDex resultante da leitura do arquivo, ao final da ordenação você deve imprimir os pokemons.

Para ler a PokeDex utilize da função "readPkDex" já disponibilizada. (Contém exemplo na Main disponibilizada).

Para imprimir um pokemon utilize a função "printPokemon" já disponibilizada. (Contém exemplo na Main disponibilizada).

Número correspondente de cada algoritmo:

1- Quick Sort

2- Heap Sort

3- TimSort

Nomes dos atributos:

"dexNumber"

"atk"

"mainType"

### **Exemplo de entrada:**

```
1
atk
dadosPokemon.csv
```

Nessa entrada o 1 indica o quickSort e portanto deve ser o algoritmo utilizado para ordenação, o "atk" foi o atributo escolhido para servir de comparação na ordenação, e "dadosPokemon.csv" o nome do arquivo a ser lido.

OBS: Os arquivos já estão no runCodes, você não precisa ter eles no seu zip. Porém, eu irei disponibilizar os arquivos para vocês poderem testar.

### **Saída:**

```

Dex#: 292 Name: Shedinja Type: bug\ghost
HP: 1 Attack: 90 Defense: 45
Speed: 40 SpcAtk: 30 SpcDef: 30

Dex#: 50 Name: Diglett Type: ground\ground
HP: 10 Attack: 55 Defense: 30
Speed: 90 SpcAtk: 35 SpcDef: 45

Dex#: 355 Name: Duskull Type: ghost
HP: 20 Attack: 40 Defense: 90
Speed: 25 SpcAtk: 30 SpcDef: 90

Dex#: 113 Name: Chansey Type: normal
HP: 250 Attack: 5 Defense: 5
Speed: 50 SpcAtk: 35 SpcDef: 105

Dex#: 242 Name: Blissey Type: normal
HP: 255 Attack: 10 Defense: 10
Speed: 55 SpcAtk: 75 SpcDef: 135

```

OBS: A função para imprimir JÁ ESTÁ FEITA, apenas utilize “printPokemon”.

**DICA:** Você pode criar uma função de comparação para o algoritmo de ordenação, e essa função que determina a qual dos atributos deve ser utilizado. Dessa forma você evita múltiplos IFs e ELSEs na função de Sort para saber por qual atributo deve ser comparado, ou ainda várias versões do mesmo algoritmo.

### Tarefa relatório:

Utilizando os arquivos “pokemonOrdem1.csv”, “pokemonOrdem2.csv” e “pokemonOrdem3.csv” disponibilizados no runCodes, que contém exatamente os mesmos Pokémon, mas em ordem inicial diferente, utilize o programa feito para ler cada um dos arquivos e ordenar o vetor de pokemons resultante da leitura pelo atributo mainType usando heap, quick e tim, para cada um dos algoritmos informe qual o nome do pokémon nos índices 0,1,2,3,4,5,6,7,8,9 para cada entrada (arquivo). Houve diferença? Se sim, por quê? Se não, poderia ter dado?

Obs: Para cada vez que for ordenar, utilize o vetor em seu estado original (Desordenado)