Recursão e Ordenação

Neste último trabalho, o aluno irá trabalhar com recursão e ordenação de listas em Python. O trabalho vale 40, e a **nota total** será composta pela soma dos seguintes itens:

- **Nota 1**: Se o trabalho funcionar corretamente sem qualquer erro de compilação ou de execução, o aluno ganha 8 pontos (ou seja, se o professor não tiver que alterar o código-fonte do aluno para chegar na resposta correta). Caso contrário, o aluno não ganha pontos neste item.
- **Nota 2**: Se a saída produzida pelo programa do aluno contiver a lista ordenada corretamente, o aluno ganha 16 pontos neste item. Caso contrário, o aluno não ganha pontos neste item.
- **Nota 3**: Se o formato de saída estiver exatamente como pedido na especificação, o aluno ganha 8 pontos neste item. Caso contrário, o aluno não ganha pontos neste item.
- **Nota 4**: Dependendo da modularização, clareza e reuso de código, o aluno pode ganhar de 0 a 8 pontos neste item.

Para estimular os alunos a criarem códigos eficientes, serão dados 2 pontos de bônus para os cinco trabalhos que executarem mais rápido (desde que estejam corretos, é claro!). As notas dos alunos serão classificadas de acordo com os seguintes critérios:

1. Pela nota total (quanto maior a nota total, melhor).
2. Caso haja empate pela nota total, os trabalhos serão ordenados pela nota 2 (quanto maior, melhor). Afinal, o professor quer premiar trabalhos com a ordenação correta.
3. Caso ainda haja empate, serão ordenados pelo tempo de execução (quanto **menor**, melhor). Afinal, o professor quer bonificar os trabalhos corretos que forem mais eficientes.
4. Se mesmo assim houver empate, as notas serão exibidas em ordem alfabética pelo nome do aluno.

Você fará um programa que ordena e classifica as notas desse próprio trabalho. Para isso, você deve ler um arquivo binário, que foi criado com auxílio do módulo `pickle`, contendo duas informações:

- Um dicionário cuja chave é o número de matrícula do aluno (número inteiro) e o conteúdo é o nome do aluno (String).
- Uma lista com as notas de cada aluno. Cada item da lista é uma tupla contendo:

- Número de matrícula do aluno
- Nota 1
- Nota 2
- Nota 3
- Nota 4
- Tempo de execução do código (em segundos)

A ordenação deve ser feita com algum dos algoritmos recursivos de ordenação ensinados nesta disciplina que executem em $O(n \cdot lg\ n)$ no caso médio, e deve ser executada uma única vez (ou seja, uma única ordenação já deve ser capaz de classificar os alunos pelos 4 critérios citados acima).

Por fim, você deve exibir o nome e a nota final dos alunos (separados por um espaço em branco), um aluno por linha, de acordo com os critérios de ordenação citados acima. A nota final do aluno será a nota total que ele obteve no trabalho acrescida de sua possível bonificação. Na hora de imprimir as notas, você deve inserir 2 pontos de bônus para os **cinco primeiros alunos** da lista. Entretanto, se houver alunos que empataram com o quinto colocado na nota total, na nota 2 e no tempo de execução, eles também receberão o bônus (afinal, seria injusto alguém perder o bônus por causa da ordem alfabética).

Como exemplo, considere os seguintes valores:

```
1  alunos = { 1 : "Bruno",  2 : "Bruna", 3 : "Maria",
2             4 : "Joao",   5 : "Jose",  6 : "Pedro",
3             7 : "Thiago", 8 : "Ana",   9 : "Rita",
4             10 : "Carol" }
5
6  notas = [ (1,  8, 16, 8, 8, 5),
7            (2,  8, 16, 8, 8, 4),
8            (3,  8, 16, 0, 8, 5),
9            (4,  8, 16, 8, 0, 5),
10           (5,  8, 0,  8, 8, 5),
11           (6,  0, 16, 0, 8, 15),
12           (7,  8, 0,  8, 4, 4),
13           (8,  8, 16, 8, 8, 15),
14           (9,  8, 16, 8, 8, 4),
15           (10, 0, 16, 8, 8, 25) ]
```

Os dados exibidos acima equivalem à tabela de notas a seguir:

Matrícula	Nome	Nota 1	Nota 2	Nota 3	Nota 4	Total	Tempo
2	Bruna	8	16	8	8	40	4
9	Rita	8	16	8	8	40	4
1	Bruno	8	16	8	8	40	5
8	Ana	8	16	8	8	40	15
4	Joao	8	16	8	0	32	5
3	Maria	8	16	0	8	32	5
10	Carol	0	16	8	8	32	25
6	Pedro	0	16	0	8	24	15
5	Jose	8	0	8	8	24	5
7	Thiago	8	0	8	4	20	4

Se ordenarmos a tabela pelos critérios do professor, teremos o seguinte resultado:

Classif.	Nome ↓	Nota 2 ↑	Total ↑	Tempo ↓
1^0	Bruna	16	42	4
	Rita	16	42	4
3^0	Bruno	16	42	5
4^0	Ana	16	42	15
5^0	Joao	16	34	5
	Maria	16	34	5
7^0	Carol	16	32	25
8^0	Pedro	16	24	15
9^0	Jose	0	24	5
10^0	Thiago	0	20	4

Para distribuir os dois pontos de bônus, notem que Joao e Maria estão empatados na quinta posição, pois possuem mesma nota total, mesma nota 2 e mesmo tempo de execução. Portanto, ambos devem receber a premiação. Com isso, a saída do programa para este exemplo deve ser **exatamente** como esta:

```
Bruna 42
Rita 42
Bruno 42
Ana 42
Joao 34
Maria 34
Carol 32
Pedro 24
Jose 24
Thiago 20
```

Observações

- O trabalho vale 40 pontos e deve ser entregue até **06 de fevereiro**.
- Trabalhos considerados **plágio** terão nota 0 para quem copiou e para quem forneceu o trabalho. Além disso, serão enviados para o Conselho de Ética.
- O código deve ser feito em Python3.
- O trabalho deve ser feito individualmente.
- Trabalhos entregues após o prazo serão automaticamente rejeitados.
- Trabalhos com erro de execução, com formato de saída incorreto, ou que não compilarem terão nota 0.
- O trabalho deve ser enviado na sala da disciplina do AVA.
- Em caso de dúvidas na especificação do trabalho ou no próprio trabalho, contate-me em hsjunioe@gmail.com