

## Prova Intermediária

### Regras da prova

- **Duração:** 180 minutos
- Cada questão possui uma pasta onde você deverá colocar suas questões.
- Dúvidas dos enunciados da prova serão resolvidos via chat.
- É permitido consultar o material da disciplina durante a prova (tudo o que estiver no repositório da disciplina e o [no site https://insper.github.io/supercomp](https://insper.github.io/supercomp). Isto inclui suas próprias soluções aos exercícios de sala de aula, **mas não inclui materiais não digitais. Consultas a outros materiais (de outros alunos ou na internet em geral) não são permitidos.**
- É possível também consultar a documentação de C++ nos sites <http://cplusplus.com/> e <https://cppreference.com>
- A prova é individual. Qualquer consulta a outras pessoas durante a prova constitui violação ao Código de Ética do Insper.

### Parte 1 - criação de algoritmos básicos

Uma empresa de entregas precisa entregar  $N$  pacotes a destinos com distância  $d_i, i = 1, \dots, N$  de sua sede. Para fazer essas entregas ela conta com  $N$  fornecedores, cada um cobrando um preço de  $p_i$  reais por quilômetro rodado. Seu trabalho será encontrar a atribuição *fornecedor-pacote* que resulte no menor custo final para realizar todas as entregas.

Seu programa deverá aceitar entradas no formato abaixo.

```
N
d1
...
dN
p1
...
pN
```

Sua saída deverá estar no formato abaixo

```
C
f1
...
fN
```

onde  $C$  é o custo final das entregas e  $fI$  é o índice do fornecedor usado para fazer a entrega de distância

### ? (1,0) Pergunta 1

Como você resolveria este problema? Descreva em alto nível um algoritmo que tenta minimizar o custo final e crie 4 pares de arquivos entrada/saída que exemplifiquem a aplicação de seu algoritmo.

### ☰ (2,0) Tarefa 1

Crie uma implementação do seu algoritmo acima. Use suas entradas/saídas para testar seu programa.

## Parte 2 - Projeto

No projeto da disciplina estamos desenvolvendo algoritmos para encontrar soluções para o problema de alocação de recursos chamado de Maximin Share (MMS), no qual devemos particionar uma coleção de  $M$  objetos indivisíveis com diferentes valores  $v_i, i = 1, 2, \dots, M$  entre  $N$  pessoas, de forma que a pessoa que tenha recebido a partição com menor valor total receba o maior valor possível.

Considere agora o problema MMS com diferentes preferências, em que o valor atribuído a cada objeto seja diferente para cada pessoa. Por exemplo, se estivermos dividindo um pacote de  $M$  balas sortidas a um grupo de  $N$  crianças, algumas crianças podem atribuir um valor maior às balas de sabor morango, enquanto outras preferem balas sabor caramelo. Assim o valor do objeto  $j$  depende da pessoa  $i$ , de forma que os valores dos objetos são definidos por uma matriz de elementos  $v_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, M$ .

Uma consequência dessa modificação do problema é que o valor da partição MMS muda dependendo da perspectiva de cada pessoa envolvida. Vejamos este exemplo do Wikipedia, em que os objetos 1 a 5 devem ser particionados entre três pessoas: Alice, George e Dina.

- Alice atribui a eles os valores 1,3,5,6,9;
- George atribui a eles os valores 1,7,2,6,8;
- Dina atribui a eles os valores 1,1,1,4,17.

Cada pessoa possui então uma partição MMS diferente:

- A de Alice é [1, 4], com valor 7;
- O de George é [1,2] ou [3,4] ou [5], todos com valor 8;
- A de Dina é [1,2,3], com valor 3.

Aqui, consideramos um particionamento como justo no sentido da MMS se ele entregar a Alice uma partição de valor de pelo menos 7, a George uma de valor de pelo menos 8 e a Dina uma de valor de pelo menos 3. Por exemplo, o particionamento [3,4] para Alice, [1,2] para George e [5] para Dina é justo, pois os valores atribuídos são 11, 8 e 17 respectivamente.

É importante notar que nem sempre podemos garantir um particionamento justo no sentido da MMS.

## (1,0) Pergunta 2

Considere a estratégia de alocação para o problema de MMS com diferentes preferências que entrega primeiro o objeto mais valioso para a pessoa 1 (sob a perspectiva dessa pessoa), depois o mais valioso para a pessoa 2 que ainda não tenha sido alocado, e assim por diante, até acabarem os objetos. Considerando Alice como a pessoa 1, George como a pessoa 2 e Dina como a pessoa 3, e considerando ainda os valores atribuídos dos objetos fornecidas no enunciado, qual é a alocação de objetos resultante dessa estratégia? Ela é justa no sentido da MMS? Qual é o valor da partição menos valiosa?

## (2,0) Tarefa 2

Implemente uma solução para o problema de alocação de objetos segundo a heurística que considera a alocação de recursos descrita na Pergunta 1. Não se preocupe em estruturar o algoritmo mais eficiente possível, desde que seja executável em tempo razoável para que possa ser testado durante a prova. Para testar sua implementação, use as entradas in1a.txt, in1b.txt, in1c.txt e in1d.txt, que possuem o seguinte formato:

```
M N
v11 v12 ... v1M
v21 v22 ... v2M
...
vN1 vN2 ... vNM
```

onde  $M$  é o número de objetos,  $N$  é o número de pessoas e  $v_{IJ}$  é o valor atribuído ao objeto  $J$  pela pessoa  $I$ .

As respostas esperadas estão em out1a.txt, out1b.txt, out1c.txt e out1d.txt que possuem o seguinte formato:

```
MMS
Objetos da pessoa 1 (ordenados pelo índice)
...
Objetos da pessoa N (ordenados pelo índice)
```

## (2,0) Tarefa 3

Implemente um programa que verifique se uma determinada alocação é justa no sentido da MMS. O programa receberá uma entrada informando a avaliação dos objetos por cada pessoa, o MMS de cada pessoa e a alocação definida. A saída deverá ser apenas SIM no caso de a alocação ser justa, e NAO no caso contrário. Use as entradas in2a-sim.txt, in2a-nao.txt, in2b-sim.txt, in2b-nao.txt que devem fornecer as saídas SIM, NAO, SIM e NAO, respectivamente. As entradas estão no formato abaixo:

```
M N
v11 v12 ... v1M
v21 v22 ... v2M
...
vN1 vN2 ... vNM

MMS_1 MMS_2 ... MMS_N
a1 a2 ... aM
```

onde  $M$  é o número de objetos,  $N$  é o número de pessoas,  $v_{IJ}$  é o valor atribuído ao objeto  $J$  pela pessoa  $I$ ,  $MMS_I$  é o valor do MMS da pessoa  $I$  e  $a_J$  é o índice da pessoa que recebeu o objeto  $J$ .

### ? (1,0) Pergunta 3

No projeto da disciplina, vocês elaboraram um programa para encontrar o máximo global para o MMS com iguais preferências, ou seja, os objetos são avaliados da mesma forma por todas as pessoas participantes do seu particionamento. Assim, a qualidade de uma possível solução era dada simplesmente pelo valor da MMS. Quando consideramos o mesmo objetivo, porém admitindo diferentes preferências, soluções que não sejam justas, porém com valor de MMS maior são de pior qualidade do que soluções justas, porém com menor MMS. Se quisermos simplesmente descartar soluções que não sejam justas, que valor deveríamos atribuir a elas? E como saber se a solução final obtida é justa?

### ☰ (2,0) Tarefa 4

Implemente um programa que resolva o problema de encontrar a MMS com diferentes preferências, e que garanta uma alocação justa no sentido da MMS. Se não for possível obter essa alocação, simplesmente informe a resposta "SEM SOLUÇÃO". Caso contrário, devolva a alocação de objetos obtida no formato pedido na Tarefa 1. Use os arquivos in3a.txt, in3b.txt, in3c.txt e in3d.txt, comparando as saídas com out3a.txt, out3b.txt, out3c.txt e out3d.txt para testar sua solução. Os arquivos de entrada estão no formato:

```
M N
v11 v12 ... v1M
v21 v22 ... v2M
...
vN1 vN2 ... vNM

MMS_1 MMS_2 ... MMS_N
```