MATA52 - Exercícios da Semana 05

- Grupo: Stromae
- Autores:
  - Bruno de Lucas Santos Barbosa (Responsável)
    - Resolvi a quarta questão implementando uma forma alternativa ao Termoo, além de auxiliar Lucas Lima no resolvimeno da terceira questão
  - Elis Marcela de Souza Alcantara
    - Resolvi a primeira questão, primeiro pensei que um desarranjo é também uma permutação, a partir disso organizei todas as permutações do objeto de entrada e filtrei somente as que se enquadravam no conceito de desarranjo (a questão da posição).
  - Monque Silva
    - Respondi a segunda questão, com base nos estudos e pesquisa sobre como usar o algoritmo de backtracking para contar todas as permutações

Instruções (não apagar)

1. **Responsável**: Após copiar este notebook, altere o nome do notebook/documentação incluindo o nome do seu grupo. Por exemplo, se você é do grupo Ouro, altere o nome do notebook para "MATA53-Semana02-Ouro.ipynb"
2. **Responsável**: Compartilhe este documento com todos os membros do grupo (para participarem da elaboração deste documento). É importante que o notebook utilizado seja o mesmo compartilhado para que os registros de participação e colaboração fiquem salvos no histórico. Sugira uma divisão justa e defina um prazo aceitável para a inserção das soluções no Colab.
3. **Responsável**: Ao concluir a atividade, compartilhe o notebook com januario.ufba@gmail.com (dando permisão para edição) e deixando o aviso de notificação marcado, para que eu receba o seu e-mail. Identificar o nome do grupo na mensagem de compartilhamento.
4. **Cada membro**: Incluia o *seu próprio nome completo* na lista de autores que auxiliaram na elaboração deste notebook. Relate brevemente a sua contribuição na solução desta lista. O responsável aparece como sendo o(a) primeiro(a) autor(a).
5. **Cada membro**: Utilize os recursos de blocos de texto e de código para adicionar as suas respostas, sem alterar os blocos de texto e código existente. Não economize, esses blocos

são de graça.

Exercícios

1. Um desarranjo é uma permutação de $p\{1, \ldots, n\}$ tal que nenhum item está na posição correta, ou seja, $p_i = i$. Implemente um código em Python que apresente todos os desarranjos de uma sequência de tamanho n.

```
import itertools

def desarranjos(p, permutacoes):
    return list((i for i in permutacoes if not any(i[j] == p[j] for j in range(len(p)

n = int(input())
p = list(map(int, input().strip().split()))[:n]

permutacoes_p = itertools.permutations(p)

print(desarranjos(p, permutacoes_p))
```

```
4
1 2 3 4
[(2, 1, 4, 3), (2, 3, 4, 1), (2, 4, 1, 3), (3, 1, 4, 2), (3, 4, 1, 2), (3, 4, 2,
```

2. Matematicamente, um multiconjunto é a generalização de um conjunto, de tal forma que permite a repetição de elementos. Por exemplo, M = {a, b, c, c, d, e, e} é um multiconjunto distinto de X = {a, b, c, d, e}, apesar de que, se M e X fossem conjuntos, teríamos M=X. Um multiconjunto de n itens pode ter menos de n! (êne fatorial) permutações. Por exemplo {1,1,2,2} tem apenas seis permutações distintas: {1, 1, 2, 2}, {1, 2, 1, 2}, {1, 2, 2, 1}, {2, 1, 1, 2}, {2, 1, 2, 1}, e {2,2,1,1}. Implemente um código em Python que apresente todas as permutações de um multiconjunto de tamanho n.

```
def permutacao(conjunto, inicio):
    if (inicio == len(conjunto)):
        print(conjunto)
    visita = set()
    for i in range(inicio, len(conjunto)):
        if conjunto[i] in visita:
              continue
        visita.add(conjunto[i])
        conjunto[inicio], conjunto[i] = conjunto[i], conjunto[inicio]
        permutacao(conjunto, inicio + 1)
        conjunto[inicio], conjunto[i] = conjunto[i], conjunto[inicio]

conjunto = [1,1,2,2]
permutacao(conjunto, 0)
```

```
[1, 1, 2, 2]
[1, 2, 1, 2]
[1, 2, 2, 1]
[2, 1, 1, 2]
[2, 1, 2, 1]
[2, 2, 1, 1]
```

Primeiro escolhemos uma combinação de um conjunto existente, dado exemplo: $[1, 1, 2, 2]$. Em seguida, juntamos o elemento inicial com todas as combinações possíveis a serem produzidas a partir de elementos que sucedem o elemento inicial.

O algoritmo backtracking implementado atua trocando os itens de volta para seu local anterior após seu valor ter sido impresso e removendo as duplicatas, de forma a obtermos todas as permutações do conjunto.

3. Crie um algoritmo que, dado n e C, imprime todas as sequências de números positivos

$x_1, x_2, \ldots, x_n$ tal que $x_1 + x_2 + \cdots x_n = C$

Acredito que o algoritmo seguiria o seguinte pseudo-algoritmo:

1. Receber as entradas C e N
2. Percorrer de 1 até C adicionando valores a uma lista

4. Considere as seguintes informações:

- P = conjunto de palavras com 5 letras.
- L = conjunto de letras de A a Z (inicialmente vazio).

Implemente um algoritmo em python que liste as palavras do conjunto P tal que nenhuma de suas letras esteja no conjunto L. (Se você estiver em um momento de inspiração, evolua a sua implementação para uma versão alternativa do jogo TERMO - https://term.ooo/)

Percebi que o Termoo funciona armazenando letras com base na palavra correta, junto com a sua posição na string, montei um mini jogo baseado em como funciona onde não é recebido um feedback do acero na posição, porém é possível ir chutando com base no feedback se a letra está ou não contida na palavra de 5 letras, recebendo um acerto final ou não ao terminar as 5 tentativas

```
#TODO: Descrever nos comentarios o desenvolvimento da questao

palavraCorreta = ["P", "A", "L", "A", "D"]
#palavra = input().upper();
# L = ["A","B","C","D","E","F",
#      "G","H","I","J","K","L",
#      "M","N","O","P","Q","R",
#      "S","U","V","W","X","Y","Z"]

L = [];
montandoPalavra = [];

for i in range(5):
    letraDigitada = input().upper();
    if(letraDigitada in palavraCorreta):
      montandoPalavra.append(letraDigitada);
      print("########A LETRA ESTÁ CONTIDA NA PALAVRA #########")
    else:
      L.append(letraDigitada);

if(montandoPalavra == palavraCorreta):
    print("PARABENS, VOCE ACERTOU A PALAVRA");
else:
    print("F")
```

```
P
########A LETRA ESTÁ CONTIDA NA PALAVRA #########
A
########A LETRA ESTÁ CONTIDA NA PALAVRA #########
L
########A LETRA ESTÁ CONTIDA NA PALAVRA #########
A
########A LETRA ESTÁ CONTIDA NA PALAVRA #########
D
```

```
########A LETRA ESTÁ CONTIDA NA PALAVRA #########
PARABENS, VOCE ACERTOU A PALAVRA
```