

**Alunos:** Samuel Augusto Meireles da Silva - 11821BSI252

Henrique Corrêa de Oliveira - 11821BSI251

# Computação Bioinspirada

Problema da Mochila Binária utilizando Algoritmo Genético

## Funcionamento do Programa:

O programa foi feito utilizando Python, se baseando em um AG genérico. Temos as variáveis de taxa de crossover, taxa de mutação, tamanho da população e quantidade de gerações. As variáveis citadas são alteradas de forma manual no programa (Imagem 1).

```
# Algoritmo Genético
tamanho_populacao = 5000
taxa_crossover = 0.7
taxa_mutacao = 1
num_geracoes = 5000
```

Imagem1

Já a inserção dos inputs é feita pela leitura de arquivos(Imagem 2),

```
def recebe_input(nome_arqv):
```

Imagem2

A definição da taxa de crossover e de mutação é a probabilidade em % delas ocorrerem, logo se tiver 0,7 de taxa de crossover, existe 70% de chance de um crossover ocorrer na execução de cada geração.

## Comparação de Resultados:

O objetivo do trabalho era comparar os resultados do problema da Mochila Binária, utilizando um Algoritmo Genético e comparando com os resultados obtidos pelo professor, utilizando o algoritmo do GRASP. Os inputs utilizados foram os mesmos para ambos algoritmos, para realizar a equiparação de resultados.

Segue a tabela de resultados do GRASP e do Algoritmo Genético:

GRASP		AG	
Input 1	31621	Input 1	31621
Input 2	67829	Input 2	67829
Input 3	143449	Input 3	
Input 4	28840	Input 4	
Input 5	15785	Input 5	
Input 6	99861	Input 6	

Input 7	1910	Input 7	
Input 8	583	Input 8	
Input 9	9581	Input 9	
Input 10	17229	Input 10	
Input 11	29965	Input 11	
Input 12	49885	Input 12	
Input 13	49398	Input 13	
Input 14	20880	Input 14	
Input 15	20676	Input 15	
Input 16	44422	Input 16	

## Resultado 1:

Para o input 1 foram utilizados os seguintes parâmetros para a execução:

```
Tamanho da População: 1500
Numero de Gerações: 5000
Taxa de Mutação: 0.25698
Taxa de crossover: 0.7
```

Os resultados obtidos foram os mesmos do GRASP:

```
Output 1 Melhor Solução:
Melhor Geração: 87
Valor total: 31621
Itens na mochila: [0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

## Resultado 2:

Para o input 2 foram utilizados os seguintes parâmetros:

```
Tamanho da População: 3000  
Numero de Gerações: 1000  
Taxa de Mutação: 1  
Taxa de crossover: 1
```

Depois de muitos testes o melhor resultado foi:

```
Output 2 Melhor Solução:  
Melhor Geração: 945  
Valor total: 67829  
Itens na mochila: [1, 0, 0, 0, 1, 0, 0, 1,  
Tamanho da População: 3000
```

O resultado também é equiparado com o GRASP