

Exemplo

Problema da Mochila booleana

Declaração de variáveis

```
int n=5; //numero de objetos
int c=60; //capacidade da mochila
int s[] = {0, 0, 0, 0, 0}; // solucao corrente
int s_star[] = {0, 0, 0, 0, 0}; // melhor solucao
int p[] = {52, 23, 35, 15, 7}; //peso de cada objeto
int b[] = {100, 60, 70, 15, 15}; //benef. de cada objeto
int fo; // funcao objetivo corrente
int excesso; // conta excesso de peso na mochila
int penalidade; //penalidade por exceder a
//capacidade da mochila (valor da soma todos itens)
int pesoObjetos; // soma dos objetos inseridos na
//solucao corrente
```

Constrói solução aleatória

```
void constroi_solucao_aleatoria(int s[], int itens)
{
    for (i=0; i<itens; i++)
        s[i] = rand()%2;//Escolhe 0 ou 1
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h> //srand

void constroi_solucão_aleatoria(int s[], int itens);

int i; //índice usado nos loops

int main()
{
    //Declaração de variáveis
    // srand(1000); //sementes de números aleatórios
    srand((unsigned) time(NULL)); // pega a hora do relógio
    constroi_solucão_aleatoria(s, n);
    printf("solução inicial aleatória:");
    imprime_solucão(s,n);
    printf("\n");
    system("pause");
    return 0;
}
```

Imprime solução

```
void imprime_solucao(int s[], int itens)
{
    for (i=0; i<itens; i++)
        printf("%d\t",s[i]);
}
```

Qualidade da solução Inicial

```
.  
. .  
//medir qualidade da solucao inicial  
    printf("penalidade: %d \n\n",50);  
    printf( "beneficio: %d \t peso na mochila: %d \t  
excesso: %d \n\n",calcula_fo(s, n, b, 50,p, n, c),  
calcula_peso_objetos(s, n, p),  
calcula_excesso(calcula_peso_objetos(s, n, p),  
c));  
    printf("\n");
```

// calcula a penalidade por exceder a capacidade da mochila

```
int calcula_penalidade(int p[], int itens)
```

```
{
```

```
    int penalidade = 0;
```

```
    for (i=0; i<itens; i++)
```

```
        penalidade += p[i];
```

```
    return penalidade;
```

```
}
```

//calcula o peso dos objetos inseridos na mochila

```
int calcula_peso_objetos(int s[], int itens, int p[])
```

```
{
```

```
    int peso = 0;
```

```
    for (i=0; i<itens; i++)
```

```
        if (s[i]) // o mesmo que (s[i]==1)
```

```
            peso += p[i];
```

```
    return peso;
```

```
}
```


//calcula o excesso de peso na mochila

```
int calcula_excesso(int peso_objetos, int  
capacidade)
```

```
{
```

```
    return MAX(0, peso_objetos - capacidade);
```

```
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h> //INT_MAX
#include <time.h>

#define MAX(x,y) ((x)<(y) ? (y) : (x))
```

Função Objetivo

Maximizar a quantidade de objetos dentro da mochila, de acordo com o benefício associado ao objeto.

$$F_o = \sum_1^n (\textit{quantidade} * \textit{benefício} - \textit{penalidade} * \textit{excesso})$$

Penalidade = parâmetro

Excesso (inviabilidade) = máximo entre zero e a diferença entre o peso dos objetos e a capacidade da mochila

$$\text{MAX}(0, \text{pesoObjetos} - c)$$

```
//calcula a funcao objetivo
```

```
int calcula_fo(int s[], int itens, int b[], int penalidade, int  
p[], int n, int c)
```

```
{
```

```
    int peso=calcula_peso_objetos(s, n, p);
```

```
    int excesso = calcula_excesso(peso, c);
```

```
    int beneficio = 0;
```

```
    for (i=0; i<itens; i++)
```

```
        if (s[i])
```

```
            beneficio = beneficio + b[i];
```

```
    return beneficio - penalidade * excesso;
```

```
}
```

Método da Descida/Subida

```
void subida(int n, int s[], int b[], int p[], int c, int penalidade, int *peso_objetos, int *fo)
{
    int pos,fo_atual,pesoObjetos, excesso;

    *fo= calcula_fo(s, n, b, penalidade,p, n , c);// vizinho anterior
    for(i=0;i<n; i++){
        for (pos=0; pos<n; pos++){
            if (i!=pos){
                troca_bit(s, pos);
                fo_atual= calcula_fo(s, n, b,penalidade,p, n , c);// vizinho atual
                if (fo_atual > *fo){
                    *fo = fo_atual;
                    printf(".....trocou [%d]\n\n", pos);
                }else{
                    troca_bit(s, pos);
                    pesoObjetos = calcula_peso_objetos(s, n, p);
                    excesso = calcula_excesso(pesoObjetos, c);
                }
            }
        }
    }
}
```

Inserere ou retira o objeto j da mochila

```
void troca_bit(int s[], int j)
{
    if (s[j])
        s[j] = 0;
    else
        s[j] = 1;
}
```

```
.  
.   
.   
#define MAX(x,y) ((x)<(y) ? (y) : (x))
```

```
int calcula_fo(int s[], int itens, int b[], int penalidade,  
int p[],int n, int c);
```

```
int calcula_penalidade(int p[], int itens);
```

```
int calcula_peso_objetos(int s[], int itens, int p[]);
```

```
int calcula_excesso(int peso_objetos, int capacidade);
```

```
void subida(int n, int s[], int b[], int p[], int c, int  
penalidade, int *peso_objetos, int *fo);
```

```
void troca_bit(int s[], int j);
```

```
.  
.   
int main()
```

```
int main()
```

```
.  
.   
.
```

```
    subida(n, s, b, p, c, penalidade, &pesoObjetos, &fo);  
    printf("\nsolucao final:");  
    imprime_solucao(s,n);  
    printf("\n\n");  
    printf( "beneficio: %d \t peso na mochila: %d \t  
excesso: %d \n\n",calcula_fo(s, n, b, 50,p, n, c),  
calcula_peso_objetos(s, n, p),  
calcula_excesso(calcula_peso_objetos(s, n, p), c));  
    system("pause");  
    return 0;  
}
```


C:\Documents and Settings\WindowsXP\Meus documentos\mochila\mochila3a.exe

solucao inicial aleatoria: 1 0 1 1 0

penalidade: 50

beneficio: -1915 peso na mochila: 102 excesso: 42

.....trocou [2]

.....trocou [3]

.....trocou [4]

solucao final: 1 0 0 0 1

beneficio: 115 peso na mochila: 59 excesso: 0

Pressione qualquer tecla para continuar. . . _

C:\Documents and Settings\WindowsXP\Meus documentos\mochila\mochila3a.exe

solucao inicial aleatoria: 0 1 0 0 1

penalidade: 50

beneficio: 75 peso na mochila: 30 excesso: 0

.....trocou [3]

solucao final: 0 1 0 1 1

beneficio: 90 peso na mochila: 45 excesso: 0

Pressione qualquer tecla para continuar. . . _

C:\Documents and Settings\WindowsXP\Meus documentos\mochila\mochila3a.exe

solucao inicial aleatoria: 1 1 1 0 0

penalidade: 50

beneficio: -2270 peso na mochila: 110 excesso: 50

.....trocou [1]

.....trocou [2]

.....trocou [4]

solucao final: 1 0 0 0 1

beneficio: 115 peso na mochila: 59 excesso: 0

Pressione qualquer tecla para continuar. . . _

F:\mochila3a.exe

solucao inicial aleatoria: 1 1 0 1 0

penalidade: 50

beneficio: -1325 peso na mochila: 90 excesso: 30

.....trocou [1]

.....trocou [3]

.....trocou [4]

solucao final: 1 0 0 0 1

beneficio: 115 peso na mochila: 59 excesso: 0

Pressione qualquer tecla para continuar. . .

C:\Documents and Settings\WindowsXP\Meus documentos\mochila\mochila3a.exe

solucao inicial aleatoria: 0 1 0 1 1

penalidade: 50

beneficio: 90 peso na mochila: 45 excesso: 0

solucao final: 0 1 0 1 1

beneficio: 90 peso na mochila: 45 excesso: 0

Pressione qualquer tecla para continuar. . . _

C:\Documents and Settings\WindowsXP\Meus documentos\mochila\mochila3a.exe

solucao inicial aleatoria: 0 1 1 0 0

penalidade: 50

beneficio: 130 peso na mochila: 58 excesso: 0

solucao final: 0 1 1 0 0

beneficio: 130 peso na mochila: 58 excesso: 0

Pressione qualquer tecla para continuar. . .