

Trabalho prático 2

Problema da Mochila

UFV – Campus Florestal

Professor:	Thais Regina de Moura Braga Silva
Disciplina:	Algoritmos e Estrutura de Dados I – CCF 211
Aluno:	Luis Viana Mudança -
Aluno:	Marcos Túlio Rodrigues Almeida - 3504

Florestal, 7 de novembro de 2019

ÍNDICE

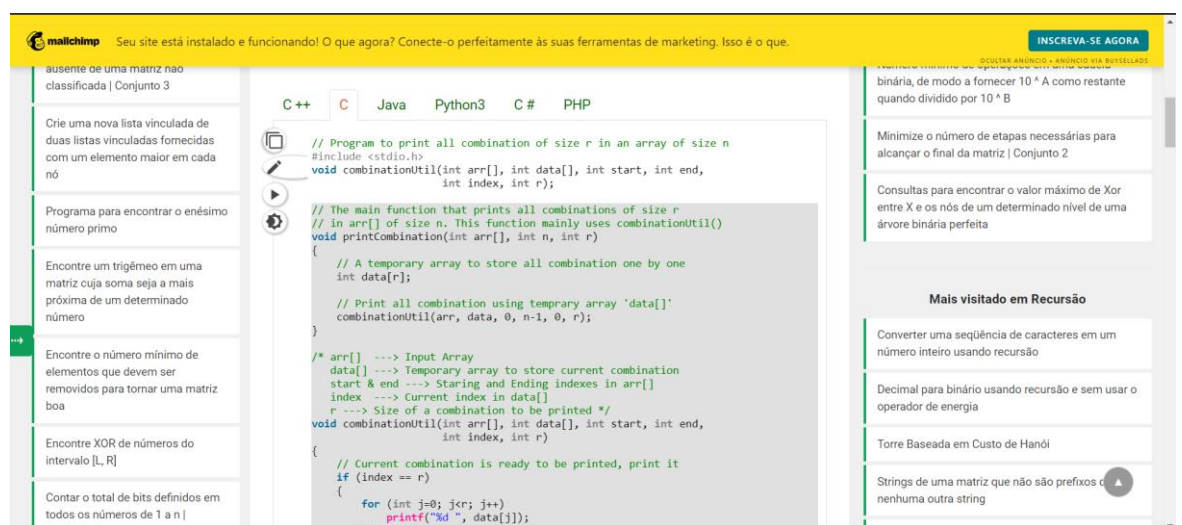
- 1. INTRODUÇÃO**
- 2. DESENVOLVIMENTO**
- 3. CONSIDERAÇÕES FINAIS**
- 4. BIBLIOGRAFIA**

1. INTRODUÇÃO

O “problema da mochila” consiste em encontrar a melhor forma de organizá-la fazendo todas as combinações sem repetições, neste trabalho poderemos perceber o quão demorado e desgastante pode ser executar essa tarefa, partindo de tamanhos maiores que 14 já podemos perceber uma pequena demora para execução do algoritmo, para valores maiores que 45 o tempo é consideravelmente alto, e para entradas de 50, 80 e 100 itens já não conseguiríamos calcular em tempo hábil.

2. DESENVOLVIMENTO

Primeiramente retiramos o algoritmo de ordenação simples do site GeekforGeeks que nos ajudou a montar a lógica para realizar a combinação de x elementos.



Site de referência para o algoritmo de combinação

Criamos um struct para armazenar os itens da mochila, como, peso e valor.

```
typedef struct
{
    int peso, valor;
} mochila;
```

Fizemos um um for para que fosse possível fazer as combinações de 1 até N.

```
for (size_t i = 0; i < n; i++)
{
    printCombination(itensMochila, n, (i + 1));
}
```

A função a seguir é responsável por realizar as diversas combinações de tamanho 1 até N de forma recursiva e além disso ela escolhe a melhor combinação de itens com maior valor agregado sem ultrapassar a capacidade máxima da mochila.

```
void combinationUtil(mochila arr[], mochila data[], int start, int end,
d,
                    int index, int r)
{
    int somaPeso = 0, somaValor = 0, j;
    mochila combinacaoTemoraria[r];

    if (index == r)
    {
        somaPeso = 0, somaValor = 0;
        for (j = 0; j < r; j++)
        {
            printf("%d %d} | ", data[j].peso, data[j].valor);
            somaPeso += data[j].peso;
            somaValor += data[j].valor;
            if (somaPeso <= CAPACIDADE && somaValor >= somaValorFinal)
            {
                somaValorFinal = somaValor;
                somaPesoFinal = somaPeso;
                if (somaValorFinal > somaValorFF)
                {
                    somaValorFF = somaValorFinal;
                    somaPesoFF = somaPesoFinal;
                }

                //printf("PESO: %d | VALOR: %d\n", somaPesoFinal, somaValorFinal);
            }
        }
        //printf("\nPESO: %d | VALOR: %d", somaPeso, somaValor);
        // combinacaoTemoraria[j] = data[j];

        printf("\n");
        return;
    }

    for (int i = start; i <= end && end - i + 1 >= r - index; i++)
```

```

{
    data[index] = arr[i];
    combinationUtil(arr, data, i + 1, end, index + 1, r);
}
}

```

• Hardware Utilizado

Exibir informações básicas sobre o computador

Edição do Windows

Windows 10 Pro

© 2018 Microsoft Corporation. Todos os direitos reservados.



Sistema

Processador: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz
 Memória instalada (RAM): 12,0 GB (utilizável: 11,9 GB)
 Tipo de sistema: Sistema Operacional de 64 bits, processador com base em x64
 Caneta e Toque: Nenhuma Entrada à Caneta ou por Toque está disponível para este vídeo

Nome do computador, domínio e configurações de grupo de trabalho

Nome do computador: boydwo
 Nome completo do computador: boydwo
 Descrição do computador:
 Grupo de trabalho: WORKGROUP

Alterar configurações

Ativação do Windows

Windows ativado [Ler os Termos de Licença para Software Microsoft](#)

ID do Produto (Product ID): 00331-10000-00001-AA809

Alterar chave do produto (Product Key)

• Resultado do tempo de execução

Teste com 5 elementos

```

melhor de todos : soma PESO: 32 | soma VALOR: 87
Tempo gasto: 0 ms.

```

Teste com 20 elementos

```
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {37 3} |  
{12 3} | {27 12} | {10 22} | {24 19} | {38 2} | {37 3} |  
{12 3} | {27 12} | {10 22} | {1 19} | {38 2} | {37 3} | {  
{12 3} | {27 12} | {24 19} | {1 19} | Seta 8 2} | {37 3} | {  
{12 3} | {10 22} | {24 19} | {1 19} | {38 2} | {37 3} | {  
{27 12} | {10 22} | {24 19} | {1 19} | {38 2} | {37 3} |  
  
melhor de todos : soma PESO: 39 | soma VALOR: 146  
{12 3} | {27 12} | {10 22} | {24 19} | {1 19} | {38 2} |  
  
melhor de todos : soma PESO: 39 | soma VALOR: 146  
Tempo gasto: 41828.1 ms.'
```

Bom para executar os outros testes foi demandado um tempo muito grande tendo em vista que deixamos executando mais de 2 horas pois o algoritmo possui comportamento assintótico $O(n!)$ e como sabemos a função fatorial é de ordem exponencial e possui um crescimento cada vez mais elevado sendo impossível esperar a solução desse problema concluindo assim que este algoritmo é dito intratável.

3. CONCLUSÃO

Aprendemos que alguns tipos de problemas são inviáveis computacionalmente dependendo do seu objetivo e o tamanho de suas entradas como é o caso do problema da mochila, que aumenta exponencialmente a cada entrada ficando maior e inviável alcançar o objetivo ótimo, com uma entrada de 200 números por exemplo o tempo calculado é indefinido, o programa iria rodar eternamente e nunca iria chegar em uma solução ótima.

BIBLIOGRAFIA

GeeksforGeeks. Disponível em: < <https://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-size-n/> >. Acesso em 01 de novembro de 2019.