

# **PROJETO DE ALGORITMOS E** **ESTRUTURA DE DADOS III**

**C204 – 2021/1**

## ○ **OBJETIVO:**

Apresentar as principais características e aplicações práticas de análise de complexidade e técnica de solução de problemas de otimização.

## ○ **CONSIDERAÇÕES INICIAIS:**

O projeto deverá ser composto de um código devidamente **comentado e indentado** e um relatório. As equipes serão de 2 ou 3 pessoas. Segue abaixo os itens necessários para a elaboração:

### **Para o Relatório:**

- Capa;
- Sumário;
- Introdução (Motivo da elaboração do próprio documento/Apresentação do projeto);
- Descrição do funcionamento do algoritmo;
- Análise de complexidade assintótica (utilize notação Big-O) e justifique-a devidamente;
- Testes e resultados, fazendo a análise do tempo de execução (utilizando o gprof) para os exemplos de entrada;
- Conclusão;
- Referências.

### **Sobre a montagem do código:**

- O código deverá ser feito utilizando a linguagem C++;
- O código deverá ser devidamente comentado e indentado, com variáveis de nome significativo;
- A solução funcionar para todos os exemplos não necessariamente significa que está correta, é recomendado que façam testes de mesa e criem mais exemplos para comprovar seu funcionamento.

## ○ **FORMA DE AVALIAÇÃO:**

O relatório deverá conter todos os itens citados anteriormente.

O valor da nota final, referente ao projeto (NL1), será composto por 35% equivalente ao relatório, 35% equivalente ao funcionamento do código e 30% equivalente a perguntas individuais que abordam os conceitos relacionados ao projeto.

## ○ O PROJETO:

O protocolo TCP é um protocolo que estabelece como deve ser feito o transporte de dados através da rede. Neste protocolo, vemos que os dados são divididos em pacotes, contendo 40 bytes de headers (para controle dos dados que serão transportados na rede) e 1460 bytes de dados propriamente ditos (para um pacote de 1500 bytes).

Imagine uma situação em que alguns arquivos devem ser enviados de um servidor para diversos aplicativos conectados à rede. Cada arquivo tem seu nível de importância, determinado neste caso por um número inteiro e que estabelece uma certa prioridade no envio. Todos eles serão enviados, porém temos a limitação de 1460 bytes por pacote, dada pelo protocolo. Para este projeto, consideraremos que cada arquivo é indivisível, ou seja, você não pode enviar parte dele em um pacote e outra parte em outro. Você deve enviar os pacotes de modo que a soma dos níveis de importância dos pacotes que vierem primeiro sejam os maiores possíveis. Veja o exemplo.

	Arquivo 0	Arquivo 1	Arquivo 2	Arquivo 3	Arquivo 4
Nível de importância	10	9	2	6	5
Tamanho do Arquivo	560 bytes	1200 bytes	400 bytes	500 bytes	500 bytes

Neste caso, o primeiro pacote a ser enviado conterá os arquivos 0, 2 e 3, cuja soma dos níveis de importância seria 18, e total de bytes 1460.

O segundo pacote a ser enviado, conterá apenas o arquivo 1, totalizando 9 de importância e tamanho 1200 bytes.

O último pacote conterá o arquivo 4, com nível de importância total 5 e peso de 500 bytes.

O seu objetivo é fazer um código que identifique quais os arquivos que devem ser enviados em cada pacote, dadas as considerações do projeto.

### Entrada

A entrada começa com um valor  $N$  ( $0 < N < 1000$ ), que indica a quantidade de arquivos a serem enviados. Logo após, seguem  $N$  linhas, cada uma contendo um inteiro  $R$  ( $0 \leq R \leq 10$ ) que indica o nível de importância do arquivo, e um inteiro  $B$  ( $0 \leq B \leq 1460$ ), que representa o tamanho em bytes deste mesmo arquivo, nesta ordem.

### Saída

A saída é composta por 2 linhas a cada pacote a ser enviado. A primeira linha indica qual o número do pacote ("Pacote 1:", "Pacote 2:", ... , "Pacote X:"), e a segunda linha indica os índices dos arquivos que serão enviados no pacote em questão.

Exemplos de entrada e saída

ENTRADA	SAÍDA
5 10 560 9 1200 2 400 6 500 5 500	Pacote 1: 0 2 3 Pacote 2: 1 Pacote 3: 4
3 10 1000 2 500 6 500	Pacote 1: 0 Pacote 2: 1 2
4 3 200 7 900 8 1200 6 540	Pacote 1: 1 3 Pacote 2: 0 2