



**1º Trabalho Prático - Dupla**

**Valor: 60 pontos (na disciplina prática)**

**Entrega: 15/05/2016**

Considere um conjunto de números inteiros  $S \subset \mathbb{Z} : |S| = n$ . Agora considere um número inteiro  $x \in \mathbb{Z}$ . O problema de combinações de elementos de um conjunto consiste em encontrar todos os subconjuntos de elementos de  $S$  tais que a soma destes elementos seja igual a  $x$ . O exemplo a seguir ilustra este problema:

$$S = \{1, 2, 3, 4, 5, 6\}, \quad x = 6$$

A solução para este problema seriam os subconjuntos

$$S_1 = \{6\}, S_2 = \{1, 5\}, S_3 = \{2, 4\}, S_4 = \{1, 2, 3\}$$

Este trabalho consiste na implementação de um programa em **C++** ou **Java** para a solução deste problema. O programa deverá receber como parâmetros um vetor de números inteiros representando um conjunto  $S$  e um número inteiro  $x$  e retornar como resultado o número de subconjuntos de  $S$  tais que seus elementos somados sejam iguais a  $x$  (não é necessário exibir cada subconjunto, apenas quantos existem).

Cada problema será representado por uma instância em arquivo texto com a seguinte estrutura:

$$x \ n \ v_1 \ v_2 \ \dots \ v_n$$

onde  $x$  é o valor a ser procurado,  $n$  é a cardinalidade do vetor e  $v_i : i = \{1, \dots, n\}$  é o valor da  $i$ -ésima célula do vetor. Lembrando que, como  $S \subset \mathbb{Z}$ , não há elementos repetidos neste vetor. Os arquivos contendo as instâncias estão disponíveis no endereço [www.dcc.ufmg.br/~fsantos/EC0036/TP1](http://www.dcc.ufmg.br/~fsantos/EC0036/TP1). Cada instância deve ser resolvida utilizando-se 1, 2, 3 e 4 threads paralelamente. Caso alguma demore mais que 3600 segundos, sua execução deve ser interrompida e o melhor resultado obtido deve ser retornado.

O programa implementado deverá usar recursos de processamento paralelo. Para isto, será usada a biblioteca **openMP** ou **Threads**. Cada instância deve ser resolvida utilizando-se 1, 2, 3 e 4 threads paralelamente. Caso alguma demore mais que 3600 segundos, sua execução deve ser interrompida e o resultado atual deverá ser exibido.

**Deverão ser enviados para o email [fernandosantos@unifei.edu.br](mailto:fernandosantos@unifei.edu.br) até as 23:59h do dia 15/05: (i) código-fonte do programa e (ii) relatório descrevendo a implementação do algoritmo e seus resultados (solução e tempo computacional para cada execução).**