



INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: FORÇA BRUTA

Prática 12

Parte 0: Preparação

Passo 1: Crie um projeto chamado **Pratica12**.

Passo 2: Adicione os arquivos que acompanham a prática ao projeto.

OBS. 1: Os arquivos inclusos se aplicam também às Prática 13 e 14.

OBS. 2: Cada arquivo tem sua própria função `main()`; configure o projeto de acordo.

subsetsum.cpp: problema do Subconjunto com Soma K (ou *Subset Sum Problem*).

subseqmax.cpp: problema da Subsequência com Soma Máxima (ou *Max. Subarray P.*).

Passo 3: Estude o código para se familiarizar com ele.

Veja o material de aula se preciso. Os problemas serão resolvidos de várias formas ao longo das próximas práticas.

Parte 1: Problema Subconjunto com Soma K

Passo 1: Implemente a função `subsetsumBF()` em **subsetsum.cpp**

A função `subsetsumBF()` deve usar um algoritmo de força bruta (*Brute Force*, BF) iterativo, como o descrito em sala. Ela deve varrer iterativamente todas as combinações de subconjuntos para determinar se algum deles tem valor igual à soma K. A complexidade dessa função é $O(2^N)$.

As funções `subsetsumBT()`, `subsetSumMemo()`, `subsetSumDP`, `subsetSumGreedy()`, serão exploradas nas próximas práticas.

Passo 2: Compile e rode a aplicação.

Faça testes modificando o tamanho do *array* para ver o impacto no tempo de execução. Verifique se as respostas são válidas (isto é, não aparece a mensagem **ERR**).

Parte 2: Problema da Subsequência com Soma Máxima

Passo 1: Implemente a função `subseqMaxBF()` em **subseqmax.cpp**:

A função `subseqMaxBF()` resolve o problema de forma ingênua (força bruta) com complexidade $O(N^2)$. Para isso, ela deve testar todas as possibilidades de início e final de sequência, encontrando a sequência com a soma máxima. Os parâmetros `ini` e `end` são usados para retornar o início e final da sequência de soma máxima, enquanto o retorno da função é a soma em si; isso vale as demais funções (`subseqMaxDC()`, `subseqMaxRec()`, `subseqMaxMemo()`, `subseqMaxPD()`) que serão exploradas nas próximas práticas.

Passo 2: Compile e rode a aplicação.

Faça testes modificando o tamanho do *array* para ver o impacto no tempo. Como a função é polinomial então é possível testar com valores grandes (acima de 100).