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: PROGRAMAÇÃO DINÂMICA

Prática 12

OBS.: Esta prática usa código da prática 10 e novos arquivos.

Parte 1: Exercitando Programação Dinâmica

Problema 1: Implemente a solução do problema Subseqmax (**subseqmax.cpp**) usando memorização (*top-down*) **ou** tabulação (*bottom-up*).

Escolha entre implementar as seguintes funções:

- `seqMaxMemo()` e `subseqMaxMemo()`: Se baseie nas versões recursivas dessas funções (`seqMax()` e `subseqMaxRec()`, dadas) e coloque memorização. Veja que a função `seqMaxMemo()` recebe dois arrays ao final (`SUM` e `INI`) que são respectivamente a memória da maior soma e da posição inicial. Esses arrays devem sem criados e inicializados em `subseqMaxMemo()` com valores -1 e passados como parâmetro para `seqMaxMemo()`. (Opcionalmente, mude para vector<int>).
- `subseqMaxPD()`: realiza o cálculo usando tabulação (sem recursão).

Desafio (Opcional): Implemente a outra opção.

Problema 2: Implemente a solução do problema Subsetsum (**subsetsum.cpp**) usando memorização (*top-down*) **ou** tabulação (*bottom-up*).

Escolha entre implementar as seguintes funções:

- `subsetSumRecMemo()`: versão com memorização da solução recursiva (Prática 10). O último parâmetro (memo) é a memória da função (um vector bidimensional), indexada pelos parâmetros n e k. Esse vector é inicializado na função `subsetSumMemo()`, já dada, que apenas chama a função `subsetSumRecMemo()`.
- `subsetSumDP()`: versão usando tabulação (sem recursão).

Desafio (Opcional): Implemente a outra opção.

Parte 2: Resolvendo problemas "reais" com Programação Dinâmica (PD)

Problema 1: Resolva o problema *Climbing Stairs* no LeetCode (LC070) usando PD.

Neste problema é pedido para calcular o número de formas distintas que podemos subir uma escada de N degraus, sendo que podemos subir 1 ou 2 degraus por vez. Por exemplo, uma escada de 3 degraus pode ser subida de 3 formas:

- 2 degraus + 1 degrau
- 1 degrau + 2 degraus
- 1 degrau + 1 degrau + 1 degrau

Implemente `climbStairs()` em **stairs.cpp** usando memorização ou tabulação.

Dica: Começa com "Fib..." e termina com "...onacci".

Problema 2: Resolva o problema *Coin Change* no LeetCode (LC322) usando PD.

Neste problema de "troco em moedas" é dado um conjunto de moedas e uma valor. Seu programa deve calcular o número mínimo de moedas que somam o valor dado. Não é necessário dizer quais são as moedas e moedas podem ser repetidas. Exemplo:

- Moedas = {1, 2, 5}
- Valor = 11
- Resposta = 3 (5 + 5 + 1)

Quando não for possível, retornar -1. A formulação recursiva é dada por:

$$Troco(valor) = \begin{cases} 0, & se\ valor == 0 \\ Infinito, & se\ valor < 0 \\ \min\limits_{m \in M}[1 + Troco(valor - m)\ ] \end{cases}$$

Implemente `coinChange()` em **change.cpp** usando memorização ou tabulação.