DCC/ICEx/UFMG
Projeto e análise de algoritmos
1º semestre de 2020

Prof. Vinicius Fernandes dos Santos
Prova - Módulo Paradigmas

Instruções gerais: A clareza e concisão das respostas também é objeto de avaliação. A **complexidade** dos algoritmos fornecidos **será levada em consideração** durante a avaliação, assim, quanto mais eficiente seu algoritmo, melhor.

1. Esta questão aborda o problema da mochila e utilizará a mesma notação vista em aula, ou seja, a entrada consiste em inteiros n e C, além de dois vetores, p e v, cada um com n elementos inteiros positivos, tais que, para todo i satisfazendo $1 \leq i \leq n$, os valores p_i e v_i correspondem ao peso e valor do i-ésimo objeto, respectivamente. Como visto em aula, deve-se determinar o maior lucro, isto é, a maior soma de valores v_i que pode ser obtida selecionando-se um subconjunto de objetos satisfazendo a restrição de que o somatório dos pesos dos objetos selecionados é no máximo C.

   Seja $M(i, j)$ o menor peso que pode ser obtido selecionando-se um subconjunto dos elementos do conjunto $\{1, \ldots, i\}$, de forma que a soma de seus valores seja j. Caso não exista tal conjunto, definimos $M(i, j)$ como ∞.

   (a) Escreva uma relação de recorrência para calcular $M(i, j)$. Não se esqueça dos casos base.

   (b) Usando a relação de recorrência do item anterior, escreva um algoritmo para resolver o problema da mochila. Analise a complexidade do seu algoritmo. A complexidade deve ser $\mathcal{O}(n \cdot V)$, onde $V = \sum_{i=1}^{n} v_i$.

   Dica: Você pode calcular $M(i, j)$ para diversos pares (i, j), armazená-los em uma matriz, e depois, de alguma forma, percorrer esta matriz para encontrar a solução.

2. Mauro está na reta final de seu doutorado. Ele quer defender o quanto antes, mas ele possui algumas pendências e sua agenda anda ocupada, com diversas atividades, desde atividades físicas até atividades sociais que ele não gostaria de faltar. As pendências que impedem Mauro de defender são referentes à escrita de k artigos científicos, cujos resultados já foram obtidos, mas que Mauro procrastinou para iniciar a escrita. Em um momento de organização nunca visto na vida de Mauro, ele conseguiu fazer um levantamento do tempo necessário para escrever os artigos necessários à sua defesa e também do tempo disponível nos próximos n dias do ano. Agora ele quer saber em que dia ele conseguirá enviar todos os artigos a seu orientador e agendar a defesa.

   Para a escrita dos artigos, Mauro quer escrevê-los na ordem, isto é, primeiro ele escreverá o artigo número 1, depois o número 2 e assim por diante. Ele sabe que para escrever o i-ésimo artigo ele precisará de a_i minutos. Além disso, ele sabe que no j-ésimo dia ele terá d_j minutos livres para se dedicar à escrita. Mauro tem uma metolodogia específica para a escrita de artigos científicos, baseada nas seguintes regras:

   (a) Os artigos devem ser escritos na ordem que ele determinou previamente;

   (b) No dia que ele termina de escrever um artigo, ele nunca começa a escrever outro;

   (c) Os artigos devem ser escritos em dias consecutivos.

   Escreva um algoritmo que determine o menor dia em que Mauro conseguiria terminar de escrever o último artigo. A entrada consiste de k, n, um vetor a com k inteiros positivos e um vetor d com n inteiros não negativos, contendo os valores descritos acima. Você pode assumir que sempre é possível escrever os k artigos nos n dias. Analise a complexidade de seu algoritmo. A complexidade do seu algoritmo deve ser tão eficiente quanto possível.

   Exemplo de entrada: $k = 2, n = 10, a = \langle 2000, 1000 \rangle, d = \langle 600, 600, 600, 600, 600, 0, 0, 600, 600, 600, 600, 600, 0, 0 \rangle$. Para esta entrada, a saída esperada é 9.

   Dicas: existe algoritmo com complexidade de tempo $\mathcal{O}(n + k)$. É possível resolver este problema com um algoritmo guloso.

3. Escreva um procedimento `produto` que receba como parâmetros um inteiro n e vetor v e determine o maior produto que pode ser obtido multiplicando-se um conjunto de um ou mais valores consecutivos das n primeiras posições de v. O procedimento deve ser recursivo e deve ter complexidade linear em n. Se desejar, o procedimento pode retornar mais de um valor (por exemplo, um par ordenado ou, de forma mais geral, uma t-upla, ou seja, uma sequência de t valores ordenados, para alguma constante t). Não se esqueça do caso base.

   Exemplo de entrada: $v = \langle -1, 0, -3, 2, -7, -9 \rangle, n = 6$. Alguns dos produtos que poderiam ser obtidos neste vetor são: -9, -1, 0, 42, 63 e 126. O maior deles é 126.

   Dica: você pode utilizar indução para resolver este problema.