



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga  
Ciência da Computação – Estrutura de Dados e Algoritmos  
Prova II – 1º/2017 – Algoritmos Gulosos e Programação Dinâmica  
Prof. Daniel Saad Nogueira Nunes

Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_

Data: 24 de junho de 2015

Duração da prova: 100 minutos
-------------------------------

Tabela de notas (uso exclusivo do professor)

Questão	Pontos	Nota
1	4	
2	4	
3	4	
Total	12	

## Observações

- Esta prova tem o total de 3 páginas (incluindo a capa) e 3 questões.
- O número total de pontos é 12.
- Certifique-se de assinar todas as folhas de resposta bem como a capa da prova.
- Leia atentamente todas as questões da prova. A interpretação do problema é crucial para o desenvolvimento correto da resposta.
- Resoluções sem justificativa não serão consideradas.
- É vedado o uso de equipamentos eletrônicos, como celulares, notebooks entre outros.
- A prova será **anulada** e medidas disciplinares serão tomadas para os alunos que “colarem” durante a avaliação.

★ Certifique-se de assinar todas as folhas de resposta.

---

### Questão 1 (4 pontos)

Pedro está querendo viajar para a cidade de sua namorada, Ana, que encontra-se do outro lado do Brasil. No decorrer da rota, existem  $n$  pontos, de modo que  $p_0$ , o primeiro ponto, é a casa de Pedro e  $p_{n-1}$ , o último ponto, é a casa de Ana. Os pontos intermediários  $p_1, \dots, p_{n-2}$  são postos de gasolina. Assuma que os pontos encontram-se em uma linha reta e que o tanque de Pedro está inicialmente cheio. Projete um algoritmo que retorne o número mínimo de paradas que Pedro deve fazer até chegar a casa de Ana dado um parâmetro  $c$ , denotando a autonomia do tanque em  $km$  e um vetor  $D$ , de modo que  $D[i]$  contém a distância entre  $p_i$  e  $p_{i+1}$  em  $km$ . Seu algoritmo deve retornar  $-1$  se Pedro não conseguir chegar a casa de Ana.

Projete o algoritmo  $VIAGEM(D, c)$  que deve possuir complexidade de tempo  $O(n)$ :

- **Entrada:** o vetor  $D[]$  e a autonomia  $c$ .
- **Saída:** o número mínimo de paradas ou  $-1$ , caso Pedro não consiga chegar à casa de Ana.

### Questão 2 (4 pontos)

O problema da menor subsequência comum, **LCS**, recebe como entrada duas *strings*  $X$  e  $Y$  e fornece como resposta o tamanho da maior subsequência comum (LCS) entre  $X$  e  $Y$ . Com base nele:

- (1 ponto) Caracterize a solução ótima por meio de uma relação de recorrência.
- (1 ponto) Dê a matriz de programação dinâmica preenchida para as *strings*  $X = ABCBDAB$  e  $Y = BDCABA$ .
- (2 pontos) Implemente o algoritmo de programação dinâmica  $LCS(X, Y)$  que recebe como entrada as duas *strings* e retorna o tamanho da maior subsequência entre elas. O algoritmo deve ter custo  $O(|X| \cdot |Y|)$ .

- **Entrada:**  $X$  e  $Y$ .
- **Saída:**  $|Z|$  onde  $Z$  é uma subsequência comum de  $X$  e  $Y$  de tamanho máximo.

### Questão 3 (4 pontos)

Tome o problema do corte de barras de aço. Dado o lucro para cada barra de tamanho  $1 \leq i \leq n$ , em  $P[i]$ , e o tamanho da barra inicial  $n$ , a saída é o máximo de lucro que pode ser obtido ao cortar (ou não) a barra em pedaços menores de tamanho inteiro. Ou seja, dada uma quantidade  $X[i] \in \mathbb{N}$  de barras de tamanho  $i$ , queremos maximizar:

$$\sum_{i=1}^n P[i] \cdot X[i]$$

Com

$$\sum_{i=1}^n X[i] \cdot i = n$$

★ Certifique-se de assinar todas as folhas de resposta.

- 
- (a) (3 pontos) Projete um algoritmo de programação dinâmica eficiente  $CUT(P, n)$  para resolver o problema do corte das barras de aço.
- **Entrada:**  $n$  o tamanho da barra inicial e  $P[]$  o lucro para cada barra de tamanho  $1 \leq i \leq n$ , de modo que  $P[i]$  indica o lucro para uma barra de tamanho  $i$ .
  - **Saída:** lucro máximo ao cortar a barra de tamanho  $n$ .
- (b) (1 ponto) Analise seu algoritmo.

I have no idols. I admire work,  
dedication and competence

---

Ayrton Senna