Insper

Programa DELTA 7o semestre - Algoritmos

Igor Montagner

2021/1

Prova DELTA de algoritmos

Esta prova contém **três questões** e tem tempo total de duração de **90 minutos**. As regras da prova são simples:

1. A prova é individual.
2. É permitido consultar qualquer material na internet. Não é permitido consultar outras pessoas.
3. Não é permitido copiar grandes trechos e/ou soluções prontas.
4. Sempre que usar uma fonte externa, adicione uma referência no seu código usando um comentário.
5. Todas as questões de implementação tem testes automatizados. A quantidade de testes que passam é mostrada ao executar o testador.
6. As questões avaliam a capacidade de desenhar uma solução para os problemas apresentados e estruturá-la usando uma linguagem de programação. Todas as questões podem ser resolvidas usando conhecimentos básicos de Linguagem *C*.
7. Em todas as questões você deverá implementar somente a função pedida. Um arquivo *.o* com uma função `main` contendo testes automatizados é fornecido para cada questão. Para testar basta executar o executável criado.

> **Importante**
>
> Um aluno estará minimamente capacitado se acertar **TODAS** as questões no tempo da prova.
>
> Para fazer a prova você precisará dos pacotes `build-essential libsystemd-dev` instalados em seu sistema.
>
> Essa prova possui um sistema de telemetria que registra quando um aluno tenta fazer uma questão. Toda informação é coletada anonimamente e é impossível associar uma tentativa de resolução a um aluno em específico.
>
> Toda a informação coletada será usada para melhorar o curso de Engenharia da Computação. **Nenhuma informação é usada para nota**.

Questão 1

Seu trabalho será criar uma função que identifica o tamanho da maior sequência estritamente crescente em um vetor. Uma sequência de números $a_i, \ldots a_k$ é crescente se $a_j < a_{j+1}$ para todo `j=i ... k-1`.

Coloque sua resposta no arquivo *questao1/solucao.c*. Sua função deverá ter assinatura `int maior_sequencia_crescente(int *vetor, int n)`. Para compilar seu programa use

```
gcc -Wall -Og solucao.c testes.o -o testes-q1 -lsystemd
```

Exemplos

O tamanho da maior sequência crescente de `3 2 1 3 5 7 9 8` é 5: ela começa no `1` e vai até o `9`.

O tamanho da maior sequência crescente de `5 6 4 7 3 8` é 2 e existe empate entre `5 6`, `4 7` e `3 8`.

Questão 2

Balancear parênteses é uma tarefa comum ao intepretar textos estruturados. Faça uma função `int parenteses_balanceados(char *texto)` que retorna `1` se os parênteses de um texto estão balanceados e `0` caso contrário.

Coloque sua resposta em *questao2/solucao.c*. Para compilar seu programa use

```
gcc -Wall -Og solucao.c testes.o -o testes-q2 -lsystemd
```

Questão 3

Você foi contratado para criar uma máquina de café que recebe pagamentos em dinheiro. Um dos pontos importantes é devolver a quantidade correta de trocos usando moedas de 50, 25, 10, 5 e 1 centavos. Você deverá fazer uma função `void troco_moedas(int reais, int centavos, int *q50, int *q25, int *q10, int *q5, int *q1)` que, dada a quantidade de reais e centavos a serem dados de troco retorne a quantidade de moedas de cada tipo a serem usadas. Você deverá usar a menor quantidade de moedas possível. Você pode supor que tem infinitas moedas a sua disposição.

Coloque sua resposta em *questao3/solucao.c*. Para compilar seu programa use

```
gcc -Wall -Og solucao.c testes.o -o testes-q3 -lsystemd
```

Exemplos

Para devolver 35 centavos de troco retornamos uma moeda de 25 e uma moeda de 10. Note que retornar 3 moedas de 10 e uma de 5 também é possível, mas usa mais moedas.