

IMD0029 - Estrutura de Dados Básicas 1 – 2017.2 – Prova 01
Prof. Eiji Adachi M. Barbosa

Nome: _____

Matrícula: _____

ANTES DE COMEÇAR A PROVA, leia atentamente as seguintes instruções:

- Esta é uma prova escrita de caráter individual e sem consultas a pessoas ou material (impresso ou eletrônico).
- A prova vale 10,0 pontos e o valor de cada questão é informado no seu enunciado. O peso desta prova é de 60% na primeira unidade.
- Preze por respostas legíveis, bem organizadas e simples.
- As respostas devem ser fornecidas preferencialmente em caneta. Respostas fornecidas a lápis serão aceitas, mas eventuais questionamentos sobre a correção não serão aceitos.
- Celulares e outros dispositivos eletrônicos devem permanecer desligados durante toda a prova.
- Desvios éticos ou de honestidade levarão à anulação da prova do candidato (nota igual a zero).

Introdução

Oh, não! Aproveitando a ausência de Mario e Luigi, que estavam de férias na Terra dos Dinossauros, Bowser dominou o Reino dos Cogumelos Algorítmicos, roubando as estrelas da aprovação e aprisionando a princesa Peach no Castelo da Recursão Infinita. Infelizmente, nossos heróis Mario e Luigi não entendem de algoritmos e não poderão salvar a princesa Peach desta vez. Você deverá resolver cada um dos problemas apresentados a seguir, para assim devolver a paz ao Reino dos Cogumelos Algorítmicos e libertar a princesa Peach. Colete o máximo de estrelas da aprovação que conseguir em cada mundo! 10 estrelas valem 1,0 ponto.

Mundo 1: (30 estrelas) O Mundo 1 é um grande deserto com um castelo em formato de pirâmide-bitônica. Logo no início deste mundo, você aprende a voar com a capa e consegue desviar dos monstros e poças de areia movediça, chegando rapidamente à pirâmide-bitônica. Ao se deparar com a pirâmide-bitônica, você não consegue encontrar uma entrada no nível térreo. Você voa ao redor da pirâmide tentando encontrar alguma entrada nas laterais, mas sem sucesso. Após sucessivos voos, você percebe que a entrada encontra-se no pico. No entanto, como a pirâmide-bitônica é muito alta, você precisa calcular exatamente a posição do pico para assim conseguir voar e aterrissar no local exato onde fica a entrada. Assuma que no Reino dos Cogumelos Algorítmicos a pirâmide-bitônica é representada como um array bitônico de inteiros sem elementos repetidos. Arrays bitônicos são aqueles que possuem uma sequência inicial crescente, seguida de uma sequência decrescente. Neste contexto, implemente uma função recursiva para encontrar o índice onde ocorre o pico de um array bitônico. Sua função deverá obrigatoriamente ter complexidade $\Theta(\lg(n))$ e seguir a assinatura:

```
int findPeak(int a[], int arraySize)
```

Obs.: Neste mundo, não existem instruções para realizar repetição, como for, while e do-while. Ou seja, você não poderá usar instruções de repetição para construir sua solução; você deverá construí-la completamente recursiva.

Mundo 2: (20 estrelas) O Mundo 2 é mal assombrado por bits-fantasmas. Neste mundo, os bits-fantasmas precisam ser colocados numa ordem correta para que se consiga abrir o portão que dá acesso à casa do fantasma Bit-Boo, o vilão deste mundo. Se capturar os bits-fantasmas já é uma tarefa difícil, colocá-los na ordem correta é pior ainda: são tantos bits-fantasmas que tentar todas as combinações possíveis é inviável! Por sorte, ao quebrar um tijolo, você encontrou um rascunho de um algoritmo de ordenação capaz de ordenar os bits-fantasmas e, assim, definir a ordem correta que eles devem ser colocados. Este algoritmo foi criado por Bit-Boo. Tentando evitar que alguém entendesse seu algoritmo de ordenação, ele ofuscou os nomes das funções no seu rascunho. O algoritmo rascunhado por Bit-Boo tem o seguinte formato:

```
void BooSort(int a[], int begin, int end)
{
    if( left < right )
    {
        int x = BooSomething(a, begin, end);
        BooSort(a, begin, x-1);
        BooSort(a, x+1, end);
    }
}
```

Ao ver o rascunho de Bit-Boo você sorriu e pensou: “Rá! Já sei que algoritmo é este! Embora as funções estejam ofuscadas, lembro bem qual algoritmo é este. Vai ser muito fácil aplicar este algoritmo e descobrir a ordem que

devo colocar os bits-fantasmas!”. Neste contexto, responda: qual é o algoritmo de ordenação que Bit-Boo rascunhou? Justifique sua resposta.

Mundo 3: (20 estrelas) No Mundo 3, o mundo das afirmações-bomba, você deverá desarmar as afirmações-bomba separando-as em verdadeiras e falsas. Desta forma, para cada uma das afirmações-bomba pelo caminho, marque V (verdadeiro) ou F (falso), **justificando sucintamente** sua resposta. Marcações de V ou F **sem justificativas não serão aceitas.**

- 1 – () O algoritmo *Merge Sort* possui complexidade assintótica $\Theta(n \cdot \log_2(n))$ para o melhor caso e $\Theta(n^2)$ para o pior caso.
- 2 – () Os algoritmos *Insertion Sort*, *Bubble Sort* e *Selection Sort* possuem a mesma complexidade assintótica para o melhor caso.
- 3 – () O *Quick Sort* possui complexidade assintótica $\Theta(n^2)$ para o pior caso, mas se selecionarmos o pivô aleatoriamente garantimos que não se cai no pior caso.
- 4 – () Analisando a complexidade assintótica para o melhor caso, o *Quick Sort* e o *Merge Sort* são mais eficientes do que o *Insertion Sort*.
- 5 – () O pior caso do *Quick Sort* é quando nas sucessivas chamadas recursivas se divide um problema de tamanho N em dois sub-problemas de tamanhos bastante similares.
- 6 – () No *Merge Sort*, é impossível dividir um problema de tamanho N em dois sub-problemas de tamanhos 0 e N-1.
- 7 – () Os algoritmos de busca sequencial e de busca binária podem ser empregados nos mesmos tipos de array.
- 8 – () Os algoritmos de busca sequencial e de busca binária possuem mesma complexidade assintótica para o pior caso.

Mundo 4: (30 estrelas) Finalmente você chegou ao mundo final, o temido vale do Bowser. Após fugir de uma horda de Koopa Troopas, as famosas tartarugas de cascos coloridos, você chegou ao portão externo do Castelo da Recursão Infinita. Para abrir este portão, você precisa de uma chave especial que você deveria ter encontrado num dos mundos passados, mas não encontrou. Por sorte, você chegou a este mundo com Yoshi-BlackSmith, o dinossauro ferreiro, que conseguirá forjar uma chave capaz de abrir o portão. Yoshi pede que você separe as moedas que você coletou ao longo da jornada de acordo com o seu metal (ouro, prata e bronze). Assim, ele poderá ir usando cada tipo de metal para partes específicas da chave. Como você está apressado, você decidiu implementar um algoritmo de ordenação eficiente. Como há uma grande repetição de moedas do mesmo metal, você decidiu implementar o *Quick Sort* que usa o algoritmo de partição que leva em consideração elementos repetidos, já que este algoritmo é mais eficiente quando há muitos elementos repetidos. Neste contexto, implemente a função de partição que agrupa os elementos iguais ao pivô no array de entrada. Por simplificação, considere que as moedas são representadas como inteiros. Sua função deverá ter complexidade $\Theta(n)$ e deverá seguir a seguinte assinatura:

```
int[] partition(int a[], int begin, int end)
```

Obs.: Diante dos poucos recursos que você e Yoshi têm à disposição, você não poderá utilizar qualquer array auxiliar para construir sua solução.