

## Prova Final

Rodrigo de Souza

14 de Setembro de 2017

1. **(1,0 ponto)** Dado um vetor  $v[0 \dots n-1]$  de números inteiros, determinar quantos números distintos há no vetor (ou seja, determinar o tamanho do conjunto de elementos do vetor). Qual é a complexidade de seu algoritmo?
2.
  - **(2,0 pontos)** Dado um vetor  $v[1 \dots n]$  de números inteiros, pode supor que dois a dois distintos por simplicidade, desejamos encontrar o  $k$ -ésimo maior elemento desse conjunto (ou seja, para  $k = 1$ , é o maior elemento, para  $k = 2$  é o segundo maior, etc.). Descreva um algoritmo que faz essa tarefa. Sua solução deve ter complexidade  $\mathcal{O}(n \lg n)$ . Você pode usar qualquer uma das estruturas de dados que estudamos, mas não pode usar ordenação (ou seja, ordenar o vetor, e pegar o elemento na  $k$ -ésima posição da esquerda para a direita).
  - **(1,0 ponto)** Descreva um algoritmo recursivo que transforma um heap em uma árvore binária de busca. Qual é a complexidade de seu algoritmo?
3. **(2,0 pontos)** Digamos que queremos implementar duas pilhas em um único vetor. Ou seja, o vetor é o espaço de memória disponível para armazenarmos essas duas pilhas. Descreva sua implementação, apresentando como as pilhas são arranjadas dentro do vetor. Uma restrição importante é que só pode ocorrer estouro de uma das pilhas quando não houver mais posições disponíveis no vetor.
4. **(2,0 pontos)** Descreva um algoritmo que recebe uma lista ligada *simples*, representando uma expressão aritmética na notação *prefixa* (operadores vem antes de operandos), e calcula o valor dessa expressão. Nessa lista, cada nó contém um operando ou um operador. Você pode usar qualquer estrutura auxiliar para responder, mas não pode copiar a expressão para um vetor e manipulá-la (ou seja, deve manipular a lista diretamente durante o cálculo). Qual é a complexidade de seu algoritmo?
5. **(2,0 pontos)**
  - Dado uma árvore binária, desejo atribuir a cada nó  $r$  (pode supor que os nós já tem um campo específico para receber esse dado) uma lista ligada com os antecessores de  $r$ , ou seja, todos os demais nós a partir dos quais podemos chegar em  $r$  com um caminho na árvore. Descreva um algoritmo recursivo que preenche a árvore com essas listas ligadas.
  - Descreva um algoritmo que calcula a profundidade de um nó dado de uma árvore binária.