

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Estrutura de dados – MATA40

Prof. Roberto Freitas Parente

Avaliação 03 — 20/02/2018

Matrícula: _____ Nome: _____

Instruções:

- a) Responda as questões com caneta esferográfica (com exceção da cor vermelha). As questões respondidas a lápis/lapiseira **não serão corrigidas** após a entrega da prova;
- b) As questões devem ser respondidas nas **folhas de respostas** entregue junto com esta avaliação;
- c) Respostas sem a devida explicação não serão aceitas;
- d) Quaisquer materiais com exceção de caneta, lápis/lapiseira e borracha estão **proibidos** e sua utilização resultará em nota **zero** na prova;
- e) As questões que forem solicitado algoritmos devem ser desenvolvidos utilizando pseudocódigo como visto em sala (sem os símbolos $\{, \}$ e $;$); e
- f) Quando for necessário utilizar um estrutura de dados é **obrigatório informar** qual **estrutura de dados** e suas operações.

Questão 1 (2.0 pontos – Cada item: 0,5). *Resolva os itens abaixo.*

- (1) *Defina um heap binário mínimo.*
- (2) *Dê dois exemplo de aplicação da estrutura de dados Heap.*
- (3) *Qual diferença de utilizar um heap binário e uma lista linear para implementar uma fila de prioridades?*
- (4) *Mostre a sequência de heaps produzida pelas operações*
“P R I O * R * * I * T * Y * * * Q U E * * * U * E”
*aplicadas a um heap **mínimo** inicialmente vazio. Uma letra significa inserção e um asterisco significa remoção do máximo. Dê a sequência de letras produzidas pelas remoções.*

Questão 2 (1.0 ponto). *Sejam U uma fila de prioridades máxima e V uma fila de prioridades mínima. Construa um algoritmo que recebe U e V como entrada e retorna uma fila de prioridade máxima com o conjunto de elementos $A = U \cup V$. Ao final do algoritmo, as filas U e V devem estar iguais como estava na entrada. Analise seu algoritmo.*

Questão 3 (3.0 pontos – 0,75 cada operação). Utilizando uma das estrutura de dados vista em sala, diferente de um heap, implemente uma fila de prioridade máxima. Implemente as seguintes operações de filas de prioridade máxima sobre um conjunto S de elementos, onde cada elemento tem um campo *chave* que deve ser utilizado como a prioridade. Analise a complexidade dos algoritmos.

- (1) *INSERIR*(S, x): insere o elemento x no conjunto S .
- (2) *MAXIMO*(S): retorna o elemento de S com maior chave
- (3) *REMOVED-MAXIMO*(S): remove e retorna o elemento de S com maior chave
- (4) *AUMENTA-CHAVE*(S, x, k): aumenta o valor da chave do elemento x para o novo valor k , onde $x.chave \leq k$ e caso contrário deve ser retornado um erro.

Questão 4 (3.0 pontos). Vimos a estrutura de dados heap binário máximo. Como poderíamos estender tal estrutura para um heap d -ário máximo? Ou seja, um heap onde cada pai tem d filhos. Defina como os valores deveriam estar organizados no heap d -ário e construa o algoritmo análogo ao *MAX-HEAPFY*(A, i) visto em sala.

Dica: Código do *Max-Heapfy*(A, i) para um heap binário máximo:

```

maior = i
se esquerda(i) ≤ A.tamheap e A[esquerda(i)] > A[maior]
    maior = esquerda(i)
se direita(i) ≤ A.tamheap e A[direita(i)] > A[maior]
    maior = direita(i)
se maior ≠ i
    troca A[i] com A[maior]
Max-heapify (A, maior)

```

Questão 5 (3.0 pontos). Dada a entrada “BBBCCCCCCCDEFFFBAAAAAAAAAACDDFB”, aplique a compactação utilizando o algoritmo de Huffman. Para tal, apresente sua árvore de Huffman, apresente a tabela contendo símbolo, frequência e código. Por fim, apresente como ficaria o código compactado e diga qual o custo de tal árvore.