

MC448: Projeto e Análise de Algoritmos I

Prof. Cid C. de Souza – 1ª Prova – (14/04/2009)

Nome:

RA:

Turma:

Observação: o peso das questões será decidido pelo docente da seguinte forma: as duas questões que você responder melhor terão peso 3 e as demais terão peso 2. Portanto, uma mesma questão pode ter peso 2 para um aluno e peso 3 para um outro aluno.

Questão	frac	Peso	Nota
1			
2			
3			
4			
Total		10,0	

Instruções:

1. A duração da prova é de 110 minutos.
2. Não é permitido usar qualquer material de consulta durante a prova.
3. **Questões mal justificadas serão consideradas erradas !**
4. O termo **ordenado** é usado nos enunciados das questões para denotar vetores ou seqüências em **ordenação não decrescente**.
5. As provas podem ser feitas a lapis porém, neste caso, ficará a critério do docente aceitar ou não eventuais pedidos de revisão de nota.
6. Use o espaço reservado para responder cada questão.
7. Use as folhas de papel almaço entregues pelo docente como rascunho.
8. Não serão consideradas respostas dadas fora dos espaços indicados.
9. Coloque o seu nome, RA e turma em **todas as folhas da prova.**
10. O uso de calculadoras está proibido durante a prova.
11. Não desgrampeie o caderno de questões.

MC448: Projeto e Análise de Algoritmos I – Prof. Cid – 1ª Prova – (14/04/2009)

Nome:

Questão	frac	Peso	Nota
---------	------	------	------

RA:

Turma:

1			
---	--	--	--

1. Considere o problema da intercalação (*merge*) de dois vetores **ordenados** A e B de n elementos (com valores quaisquer) para obter um novo vetor **ordenado** C de tamanho $2n$ contendo todos os elementos de A e de B .

Zé e Mané são dois alunos de MC448 muito aplicados. Zé disse a Mané que ele encontrou um algoritmo baseado em comparações entre elementos dos vetores que resolve o problema da intercalação em tempo $O(n^k)$ para algum $0 < k < 1$. Mané disse que isto era impossível.

Mané está com a razão ou não ? Justifique cuidadosamente a sua resposta.



MC448: Projeto e Análise de Algoritmos I – Prof. Cid – 1ª Prova – (14/04/2009)

Nome: _____

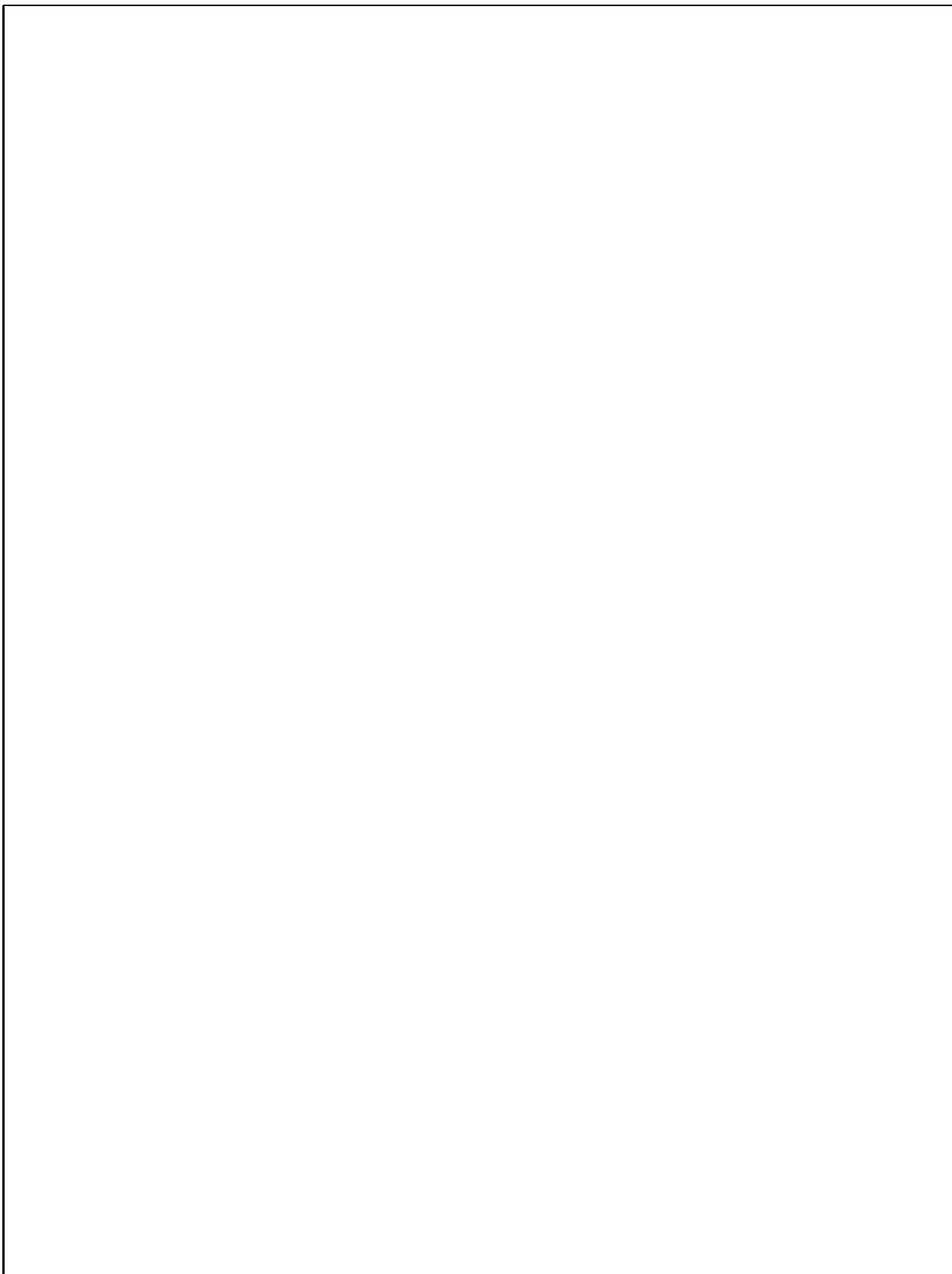
RA: _____

Turma: _____

Questão	frac	Peso	Nota
2			

2. Seja $A[1 \dots n]$ um vetor com n elementos **distintos**. Se $i < j$ e $A[i] > A[j]$ então o par (i, j) é dito ser uma **inversão**. Responda os itens a seguir.

- (a) Encontre uma **fórmula fechada** em n para o número máximo de inversões que pode existir em A . Explique cuidadosamente como você chegou a essa fórmula e diga qual seria a característica do vetor A para o qual este máximo é atingido.
- (b) Qual a relação entre a complexidade de execução do algoritmo de ordenação por inserção (INSERTION SORT) e o número de inversões do vetor de entrada A ? Justifique cuidadosamente a sua resposta.
- (c) Usando **indução matemática**, projete um algoritmo que calcula o número de inversões no vetor A e cuja complexidade de **pior caso** seja $\Theta(n \log n)$. Você não precisa dar um pseudo-código do algoritmo mas deve explicar com cuidado os argumentos da prova por indução.
[Dica: pense no algoritmo de ordenação por intercalação (MERGE SORT).]





MC448: Projeto e Análise de Algoritmos I – Prof. Cid – 1ª Prova – (14/04/2009)

Nome:	
RA:	Turma:

Questão	frac	Peso	Nota
3			

3. Abaixo encontra-se um pseudo-código do algoritmo AjustaHeap vista em sala de aula. Seja $T(n)$ a complexidade de **pior caso** deste algoritmo (n é o tamanho do *heap*). Uma vez que o algoritmo é recursivo, Ana Saab Tudor, ex-aluna de MC448, descreveu esta complexidade através da recorrência

$$T(n) \leq T\left(\frac{2n}{3}\right) + d,$$

sendo d uma constante positiva. Responda os itens abaixo:

- Explique detalhadamente o significado de cada um dos termos da fórmula proposta por Ana.
- Resolva esta recorrência usando o Teorema Master para mostrar que $T(n) \in O(g(n))$, sendo $g(n)$ a função vista em aula para a complexidade do AjustaHeap. Deixe claro qual dos casos do Teorema esta recorrência se encaixa e porquê.
- Demonstre o resultado acima usando o método de substituição (indução matemática). Para isto, considere que $T(1) = 0$.

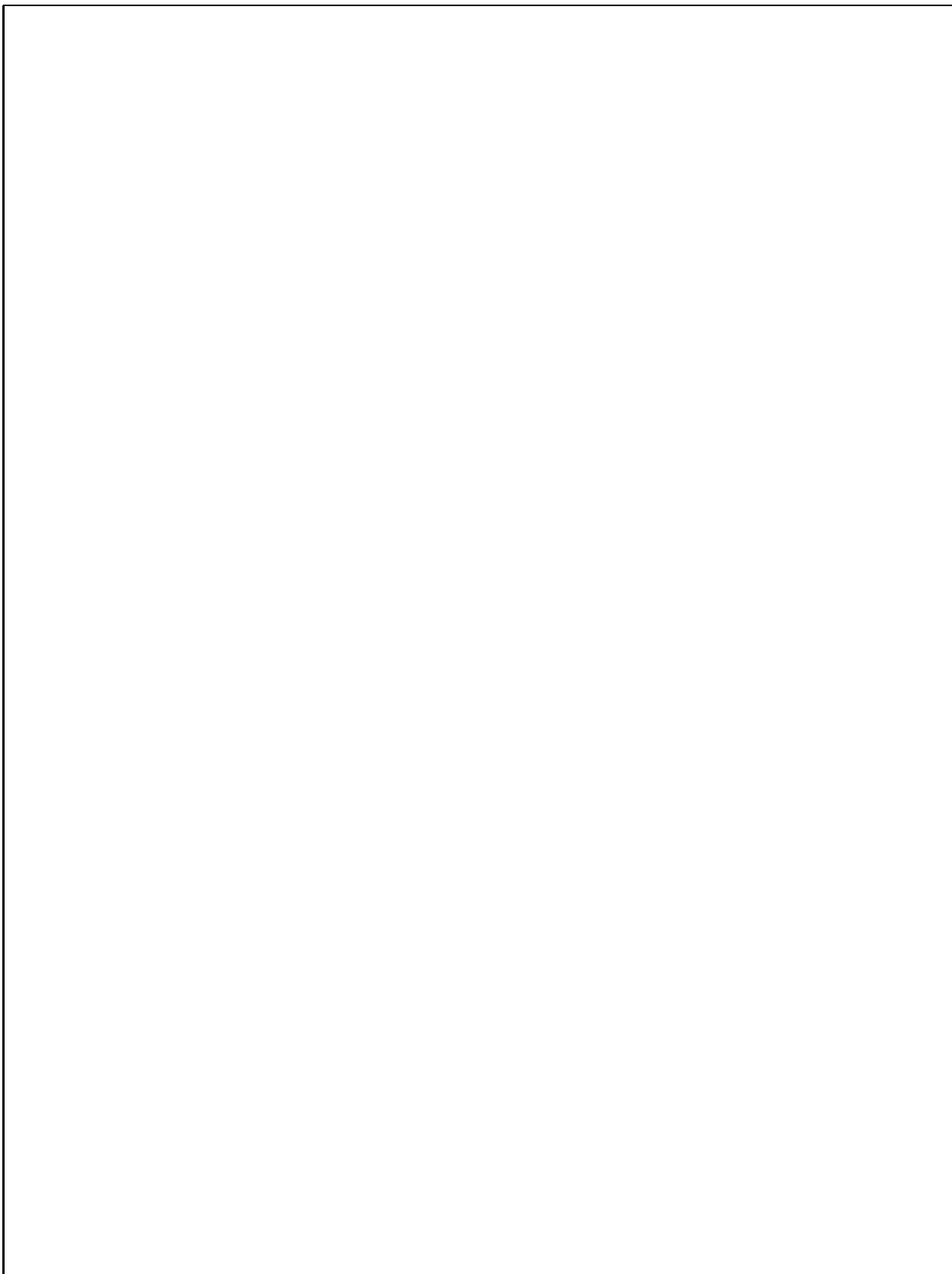
AjustaHeap(A, i, n)

▷ **Entrada:** Vetor A de n números inteiros com estrutura de heap, exceto, talvez, pela subárvore de raiz i .

▷ **Saída:** Vetor A com estrutura de heap.

- se** $2i \leq n$ e $A[2i] > A[i]$
- então** maximo $\leftarrow 2i$ **se não** maximo $\leftarrow i$
- se** $2i + 1 \leq n$ e $A[2i + 1] > A[\text{maximo}]$
- então** maximo $\leftarrow 2i + 1$
- se** maximo $\neq i$ **então**
- $t \leftarrow A[\text{maximo}]; \quad A[\text{maximo}] \leftarrow A[i]; \quad A[i] \leftarrow t;$
- AjustaHeap(A, maximo, n)

--





MC448: Projeto e Análise de Algoritmos I – Prof. Cid – 1ª Prova – (14/04/2009)

Nome: _____

RA: _____

Turma: _____

Questão	frac	Peso	Nota
4			

4. Seja A um vetor ordenado de n números **inteiros** e **distintos**. Usando **indução matemática**, projete um algoritmo de complexidade $o(n)$ (o minúsculo !) que determina se existe um índice $i \in \{1, \dots, n\}$ tal que $A[i] = i$ (ou seja, um elemento de A que seja igual ao seu índice).

Além da prova por indução, sua resposta deverá ter um pseudo-código do algoritmo, a fórmula de recorrência que descreve a sua complexidade e a resolução desta recorrência mostrando que ela é de fato $o(n)$. Se usar o Teorema Master, diga em qual dos casos a recorrência se encaixa e porquê.

