MAC 338 – Anáise de Algoritmos
PRIMEIRO SEMESTRE DE 2011
Segunda Prova – 13 de maio

Nome do aluno: ______________________________ **Curso**: ____________

Assinatura: ____

No. USP: ____________________ **Professor**: ______________________

Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota!
4. A prova consta de 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. A prova é sem consulta.

Não escrever nesta parte da folha

Questão	Nota	Observação
1		
2		
3		
Total		

Boa prova!

1. [1,0+1,0+1,0 pontos]
   Suponha que temos que ordenar um vetor com n registros e que a chave de cada registro (o campo que deve ser usado para a ordenação) tem um valor 0 ou 1. Um algoritmo para ordenar esse vetor pode ter um subconjunto das seguintes características:

   (i) o algoritmo consome tempo $\mathrm{O}(n)$;

   (ii) o algoritmo é estável;

   (iii) o algoritmo ordena no lugar (ou seja, não usa vetor auxiliar para fazer a ordenação; apenas eventualmente um número constante de variáveis simples).

   (a) Dê um algoritmo que tenha as características (i) e (ii).

   (b) Dê um algoritmo que tenha as características (i) e (iii).

   (c) Dê um algoritmo que tenha as características (ii) e (iii).

2. [1,0+0,5+1,5 pontos] (Esta questão é uma modificação de um dos exercícios de uma das listas.)

   Seja $x_1, x_2, \ldots, x_n$ uma seqüência de números, onde n é par. Um *pareamento* de $x_1, x_2, \ldots, x_n$ é uma partição do (multi)conjunto $\{x_1, x_2, \ldots, x_n\}$ em pares. Se P é um pareamento de $x_1, x_2, \ldots, x_n$, então a *altura* de P é o valor $\max\{x_i + x_j : \{x_i, x_j\}$ é um par de $P\}$. Um pareamento de $x_1, x_2, \ldots, x_n$ é *ótimo* se tem altura mínima.

   (a) Escreva (em pseudo-código, como nas aulas) um algoritmo que, dado n e um vetor $x[1 \mathinner{..} n]$ com n números, encontre e devolva um pareamento ótimo de $x[1], x[2], \ldots, x[n]$. Seu algoritmo deve consumir tempo $\mathrm{O}(n \lg n)$.

   (b) Analise o consumo de tempo do seu algoritmo, concluindo que de fato é $\mathrm{O}(n \lg n)$.

   (c) Prove que ele de fato produz um pareamento ótimo, ou seja, que nenhum emparelhamento pode ter uma altura menor que a do emparelhamento produzido pelo seu algoritmo.

3. [1,5+1,0+1,0+0,5 pontos]

Você conhece a tartaruga Yertle?

O trono de Yertle é composto de uma pilha de tartarugas, conforme a figura. Cada tartaruga, ao se alistar para fazer parte do trono, fornece o seu peso e a sua força. A sua missão é determinar uma pilha de tartarugas, dentre as alistadas, o maior possível, em que nenhuma tartaruga quebre por entrar na pilha.

O peso de cada tartaruga é dado em gramas, e a sua força é em gramas também. A força de uma tartaruga indica quanto peso, incluindo o seu, ela aguenta. Ou seja, uma tartaruga de peso 300 gramas que tem força de 1000 gramas pode carregar 700 gramas de tartarugas sobre ela. Numa pilha válida de tartarugas, cada tartaruga tem força para sustentar as tartarugas que estão acima dela na pilha.

Considere as três possíveis maneiras de ordenarmos as tartarugas:

(i) em ordem decrescente de peso;

(ii) em ordem decrescente de força;

(iii) em ordem decrescente de força menos peso.

(a) Para cada uma destas três ordens, prove ou dê um contra-exemplo para a seguinte afirmação: qualquer pilha válida de tartarugas pode ser reorganizada para que as tartarugas apareçam nesta ordem, de baixo para cima, e a pilha continue válida. *Dica:* a afirmação é verdadeira para apenas uma das ordens.

(b) Seja n o número de tartarugas, $p[1..n]$ o peso das n tartarugas e $f[1..n]$ a força das n tartarugas. Suponha que as tartarugas estão ordenadas na ordem correta (a que funciona das três acima) e que todas tem peso maior que zero. Seja $c[i, w]$ o número de tartarugas na maior pilha válida de tartarugas composta de tartarugas de 1 a i com peso igual a w. (Se não houver pilha válida, deixe $c[i, w] = -1$.) Escreva uma recorrência válida para $c[i, x]$. Não esqueça de definir o caso base da recorrência! Explique porque sua recorrência está correta, ou seja, prove (indutivamente) que $c[i, w]$ é o número de tartarugas na maior pilha válida de tartarugas composta de tartarugas de 1 a i com peso igual a w

(c) A partir da sua recorrência, escreva (em pseudo-código, como nas aulas) um algoritmo de programação dinâmica que, dados n, p e f, representando os dados de n tartarugas, determine e devolva o número de tartarugas em uma pilha válida máxima formada com estas tartarugas. O seu algoritmo deve claramente corresponder à sua recorrência.

(d) Analise o consumo de tempo do seu algoritmo em função de n.