

MAC 5711 – Análise de Algoritmos
PRIMEIRO SEMESTRE DE 1999
Segunda Prova – 7 de maio de 1999

Instruções

- Esta prova consiste de duas partes. A primeira, você deve fazer agora, em classe, e tem 2 horas e meia de duração. A segunda parte você deve entregar na Secretaria do Departamento de Ciência da Computação (sala 256-A) até segunda, dia 10 de março de 1999, às 12 horas.
- Para a segunda parte, você pode usar as suas notas de aula e o livro texto (T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms, MIT Press & McGraw-Hill, 1992). **Você não deve discutir as questões da prova com ninguém.**
- A primeira parte da prova é sem consulta e você não deve ficar nervoso! 😊
- Na segunda parte da prova, você não precisa resolver novamente todos os exercícios da prova. Resolva apenas aqueles que você gostaria de acrescentar ou corrigir algo que você fez na primeira parte.
- Os pesos que serão dados as duas partes da prova ainda não estão definidos, porém a primeira parte deve receber um peso maior ou igual ao da segunda parte.

1. (**Valor: 2.0 pontos**) Nesta questão, você deve **descrever** (e não escrever) para cada item um procedimento que recebe um inteiro positivo n e dois vetores $Nome$ e $Nota$. O vetor $Nome$ contém os nomes dos alunos de uma disciplina, e o vetor $Nota$ contém a nota de cada aluno na disciplina. A nota é uma letra entre 'A' e 'D'. (Resumindo, $Nota[i]$ é o conceito ('A', 'B', 'C' ou 'D') obtido pelo aluno cujo nome está em $Nome[i]$.)
O seu procedimento deve rearranjar os dois vetores de forma a deixar primeiramente os alunos cujo conceito é 'A', em seguida os alunos com conceito 'B', depois os alunos com conceito 'C' e finalmente os alunos com conceito 'D'. Os alunos que receberam um mesmo conceito devem estar ordenados por ordem alfabética.
Descreva (e não escreva) um procedimento para fazer isso o mais eficiente possível para cada um dos seguintes casos. Justifique a sua resposta.
 - (a) O vetor $Nome$ já está ordenado em ordem alfabética.
 - (b) Nenhum dos dois vetores está ordenado.
2. (**Valor: 2.0 pontos**) Suponha que se quer determinar os k menores elementos de um vetor com n elementos distintos, e não se está interessado na ordem relativa entre estes elementos. Descreva, se possível, um algoritmo que execute em tempo linear para cada um dos seguintes casos. Justifique a sua resposta.
 - (a) k é constante.
 - (b) k é parte da entrada do algoritmo, ou seja, não é constante.
3. (**Valor: 2.0 pontos**)
 - (a) Uma seqüência de n operações é executada em uma estrutura de dados. A i -ésima operação custa i se i é uma potência de 3, e 2 caso contrário. Determine o tempo amortizado por operação. Justifique a sua resposta.
 - (b) Considere as seguintes operações sobre uma pilha P . As tradicionais $empilha(P, x)$ e $desempilha(P)$, e a $multidesempilha(P, k)$ explicada a seguir. A operação $multidesempilha(P, k)$ consiste em desempilhar q elementos da pilha, onde q é o mínimo entre k e o número de elementos na pilha imediatamente antes da execução do $multidesempilha(P, k)$. Considere que uma seqüência de n destas operações é executada em uma pilha inicialmente vazia.
Análise o tempo de execução no pior caso de cada uma destas três operações. Descreva uma análise amortizada para estas operações, usando o método de créditos. Conclua qual é o custo amortizado por operação de acordo com a sua análise.

4. (**Valor: 2.0 pontos**) Considere a representação de uma partição do universo $\{1, 2, \dots, 8\}$ usando florestas disjuntas (como descrito em aula) com as heurísticas da união por *rank* e compressão de caminhos. Mostre como pode estar a floresta e a função *rank* em cada nó após **cada uma** das seguintes operações, se inicialmente cada elemento está em um conjunto unitário (com rank 0).
- (a) `union(1,2)`
 - (b) `union(3,4)`
 - (c) `union(5,6)`
 - (d) `union(1,3)`
 - (e) `find(2)`
 - (f) `union(7,8)`
 - (g) `union(5,7)`
 - (h) `union(1,5)`
 - (i) `find(3)`
 - (j) `find(7)`
5. (**Valor: 2.0 pontos**) Seja σ uma seqüência de *unions* e *finds* nos quais todos os *unions* ocorrem antes dos *finds*. Prove que, com a implementação de conjuntos disjuntos por florestas disjuntas com união por *rank* e compressão de caminhos, qualquer seqüência de m operações é executada em tempo $O(m)$. Assuma que os *unions* são todos chamados a partir das raízes (ou seja, que nenhum *find* é (implicitamente) necessário antes da chamada de um *union*).

MAC 5711 – Análise de Algoritmos
PRIMEIRO SEMESTRE DE 1999
Questionário de Avaliação

Você deve entregar este questionário respondido junto com a segunda parte da sua prova. Use o verso desta página, se precisar.

1. Esqueça por um minuto a sua nota e a obrigatoriedade desta disciplina. Você diria que
 - gosta da disciplina apesar da sua dificuldade.
 - gosta apesar da sua facilidade. 😊
 - não gosta mas acha a disciplina importante para a sua formação.
 - gosta.
 - gosta muito.
 - não gosta.
 - não gosta e acha inútil.
 - outro:
2. Quantas horas por semana você gasta estudando para esta disciplina fora da sala de aula?
3. Quantas vezes você já começou a fazer esta disciplina? Você desistiu logo no início do curso? Fez até o fim mas não passou?
4. Quantas disciplinas, além desta, você está cursando este semestre? (Diga quantas você começou a fazer, de quantas você desistiu, e porquê.)
5. Tendo em mente que você alguma hora vai escrever uma dissertação de mestrado (ou de doutorado), você acha que esta disciplina lhe ajuda significativamente a melhorar em algum aspecto?
6. Você acha que deveriam haver menos ou mais listas de exercícios? Há algo sobre as listas de exercícios que você acha que deveria ser diferente para garantir melhor o seu aprendizado?
7. Há alguma coisa que você acha que possa ser feita para facilitar o seu aprendizado desta disciplina?
8. Se você tivesse que sugerir um conjunto de três disciplinas centrais para um mestrado em Ciência da Computação, quais disciplinas você sugeriria?