MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Programa de Pós-Graduação em Ciência da Computação
Disciplina: Projeto e Análise de Algoritmos
Professor: David Menotti Gomes

Trabalho Prático 1

Paradigmas de Projeto de Algoritmos

Para cada um dos exercícios propostos abaixo, além da implementação de algoritmos exigida, deve-se:

- Executar os algoritmos para 5 instâncias de tamanhos e valores diferentes. Procure utilizar instâncias que levem os algoritmos a exceder os limites de espaço (memória física de seu computador) e tempo (mais de 10 minutos).
- Comparar os resultados obtidos analisando-os e discursando sobre eles.

Valor: 1,0 pontos (10% da nota total) Data de Entrega: 23/05/2011

1. (Divisão-e-Conquista) Considere A e B matrizes quadradas de ordem $n \times n$ compostas de elementos a_{ij} e b_{ij}, respectivamente, com $i, j = 1, 2, ..., n$. No produto $C = A.B$, pode-se definir o elemento c_{ij} de C, para todos $i, j = 1, 2, 3, ..., n$, como

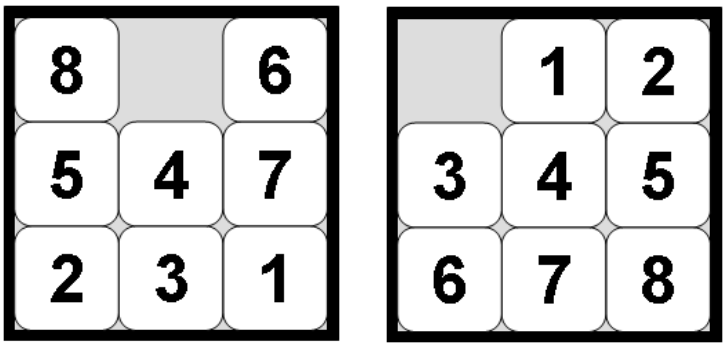
$$c_{ij} = \sum_{k=1}^{n} a_{ik}.b_{kj}.$$

   Para calcular a matriz produto C, deve-se calcular n^2 elementos cada um com um custo de $\Theta(n)$ somas e produtos. Portanto, usando-se este procedimento tem-se $\Theta(n^3)$ para o computo da matriz produto C.

   Utilizando-se do paradigma de **divisão-e-conquista**, é possível realizar o cálculo do produto de duas matrizes quadradas em $o(n^3)$. Apresente um algoritmo com tal complexidade e implemente-o.

2. (Tentativa e Erro) Considere o problema do quebra-cabeça de 8 peças (8-*Puzzle*), em uma grade para 9 lugares, onde as 8 peças estão numeradas de 1 a 8 e dispostas sobre essa grade com uma única posição vazia, e as peças só podem ser movimentadas em sentido horizontal ou vertical em rumo a posição vazia. O objetivo do problema é colocar todas as peças em ordem, como ilustra a Figura 1.

   Então, solicita-se que seja apresentado um algoritmo que utiliza o paradigma de **tentativa e erro** (ou *Backtracking*) para resolver o problema 8-Puzzle. Após apresentado, o algoritmo deve ser implementado.



(a) (b)

Figura 1: 8-Puzzle: (a) Peças desorganizadas. (b) Peças ordenadas.

3. (Programação Dinâmica) Considere o problema de **cortar um bastão de aço** de comprimento n e uma dada tabela de preços de bastões p_i de comprimento $i = 1, 2, ..., n$, e determinar a máxima receita r_n que se pode obter cortando o bastão de aço de comprimento n e vendendo seus pedaços. Para ilustrar considere a seguinte tabela.

Comprimento i	1	2	3	4	5	6	7	8	9	10
Preço p_i	1	5	8	9	10	17	17	20	24	25

   Para este caso, $n = 10$, tem-se que a receita máxima é obtida cortando-se o bastão de comprimento 10, em três bastões menores de comprimento 6, 2 e 2, obtendo-se uma receita de 27. Implemente um algoritmo usando o paradigma de **programação dinâmica** para calcular os cortes em um bastão de aço de comprimento n com uma data tabela p_i, com $i = 1, 2, ..., n$ que produzirão a receita ótima (máxima).

4. (Algoritmos Gulosos) Considere o problema inteiro da mochila:

   - Um ladrão acha n itens numa loja.
   - Item i vale v_i unidades (dinheiro, *e.g.*, R$, US$, etc).
   - Item i pesa p_i unidades (kg, etc).
   - v_i e p_i são inteiros.
   - Consegue carregar P unidades no máximo.
   - Deseja carregar a "carga" mais valiosa.

   Implemente três algoritmos usando abordagens **gulosas** para calcular os itens que podem ser carregados na mochila respeitando a capacidade P.

5. (Algoritmos Aproximados) Suponha que X representa um conjunto de habilidades que são necessárias para se realizar uma tarefa na Lua. Suponha que existem várias equipes S de astronautas disponíveis para trabalhar nesta tarefa. As famílias de possíveis equipes são representadas por $\mathcal{F}$, e pode-se dizer que $X = \bigcup_{S \in \mathcal{F}} S$. Considere que astronautas podem fazer parte de mais de uma equipe ao mesmo tempo.

Deseja-se formar um comitê (conjunto de equipes) com o menor número de *equipes* possível, de tal forma que pelo menos um membro de uma das equipes possua uma dada habilidade em X. Considere que este comitê mínimo seja representado por $\mathcal{C}$, então pode-se escrever: $X = \bigcup_{S \in \mathcal{C}} S$.

Para ilustrar, considere $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $S_1 = \{1, 2, 5, 6, 9, 10\}$, $S_2 = \{6, 7, 10, 11\}$, $S_3 = \{1, 2, 3, 4\}$, $S_4 = \{3, 5, 6, 7, 8\}$, $S_5 = \{9, 10, 11, 12\}$, $S_6 = \{4, 8\}$, e $\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$. O comitê mínimo $\mathcal{C}$ que cobre X é composto pelos conjuntos S_3, S_4 e S_5, *i.e.*, $\mathcal{C} = \{S_3, S_4, S_5\}$ e tem tamanho 3.

Implemente um algoritmo **aproximado** para resolver este problema que também é conhecimdo como **problema de cobertura de conjuntos**.

O que deve ser entregue

- Código fonte dos programas em C ou C++ (bem identado e comentado).
- Documentação do trabalho.

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição de cada problema a ser resolvido.
2. Implementação: descrição sobre a implementação do programa. Muito importante: os códigos utilizados nas implementações devem ser inseridos na documentação.
3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementadas.
4. Análise de Resultados: comparar os dados obtidos e discussar sobre estes.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sítio da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto onde é utilizada.
7. Em LaTeX: A documentação deve ser elaborada obrigatoriamente em LaTeX. Veja modelo de como fazer o trabalho em latex: `http://www.decom.ufop.br/menotti/paa111/tps/modelo.zip`
8. Formato final: mandatoriamente em PDF (`http://www.pdf995.com/`).

Como deve ser feita a entrega

A entrega DEVE ser feita via Moodle (`www.decom.ufop.br/moodle`) na forma de um único arquivo zipado, contendo o código fonte, arquivos diversos e a documentação. Também deve ser entregue a documentação impressa[1] na próxima aula teórica após a data de entrega do trabalho.

[1] Recomenda-se utilizar impressão frente-e-verso para economia - você estará ajudando a salvar o planeta e as costas do professor, também!

Comentários Gerais

- A maioria destes problemas foram extraídos de [2, 1, 3] e as vezes de alguma forma alterados;
- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero. Devido a recorrentes problemas com cópias de trabalhos (plágios), os autores de trabalhos copiados também terão a maior nota dentre os testes teóricos levada a zero, como forma de punição e coação ao plágio acadêmico;
- Trabalhos entregues em atraso terão descontados 0,1 pontos por hora;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.

Referências

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press/McGraw-Hill, 2nd edition, 2001.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press/McGraw-Hill, 3rd edition, 2009.

[3] M. J. Quin. *Parallel Computing Theory and Practice*. McGraw-Hill, 1994. ISBN-10: 0071138005.