PUC GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
CMP1065 Projeto e Análise de Algoritmos I

Marco A. F. Menezes

AULA 11

Referências: esta aula está baseada nos seguintes livros,

1. Ruy Campello e Nelson Maculan. Algoritmos e Heurísticas: desenvolvimento e avaliação de performance. Niterói, RJ: EDUFF, 1994.

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein. Algoritmos: teoria e prática. Tradução da segunda edição americana: Vandenberg D. de Souza. Rio de Janeiro: Campus, 2002.

3. Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani. Algoritmos. Tradução: Guilherme A. Pinto. São Paulo: McGraw-Hill, 2009.

4. Kenneth H. Rosen. Matemática Discreta e Suas Aplicações. Tradução: Helena Castro e João Guilherme Giudice. Sexta edição, São Paulo: McGraw-Hill, 2009.

5. Jayme L. Szwarcfiter. Grafos e algoritmos computacionais. Rio de Janeiro: Campus, 1984.

6. Laira V. Toscani e Paulo A. S. Veloso. Complexidade de Algoritmos. Série Livros Didáticos, Número 13, Instituto de Informática da UFRGS. Segunda edição. Porto Alegre: editora Sagra Luzzatto, 2005.

7. Nivio Ziviani. Projeto de Algoritmos: com implementações em PASCAL e C. Terceira edição revista e ampliada. São Paulo: Cengage Learning, 2015.

Aula anterior

2.2 Programação dinâmica

Programação dinâmica Seja P um problema dado. Às vezes, é possível decompor P em um certo número de subproblema P' de natureza igual ou semelhante a P, de tal modo que, resolvendo cada um desses subproblemas, obtém-se uma solução para P. Naturalmente, cada um dos subproblemas P' deve ser de tamanho menor do que P. Frequentemente, esses subproblemas não são necessariamente diferentes. Isto é, a solução de P pode depender de m_1 problemas P'_1, m_2 problemas P'_2, e assim por diante. Para obter uma solução para P, nesse caso, seria obviamente mais interessante resolver uma única vez, ao invés de m_i vezes cada um desses subproblemas P'_i. Para tanto, uma ideia seria resolver o subproblema P'_i, na primeira vez que esse fosse considerado, armazenando-se o resultado em uma tabela. Assim sendo, em todas as ocasiões subsequentes em que fosse necessário resolver novamente P'_i, bastaria uma consulta à tabela em questão, para obter o resultado desejado. Essa técnica, essencialmente, constitui o que se denomina (projeto de algoritmo) programação dinâmica.

A estratégia de programação dinâmica: constitui um processo de recursão no qual dois procedimentos recursivos idênticos são computados uma única vez.

Sequência de Fibonacci - Leonardo Pisano (filho de Bonacci): $(0, 1, 1, 2, 3, 5, 8, \ldots)$. Quantos pares de coelhos existirão após um ano, iniciando com um único par (macho e fêmea) se todo mês cada par produz um novo par, o qual torna-se produtivo do segundo mês em diante se não houver nenhuma morte? Sendo a_n o número de pares de coelhos após n meses, e considerando que a produtividade de um par novo se dará do segundo mês em diante, a seguinte relação de recorrência modela o processo de crescimento da população de coelhos:

$$a_n = a_{n-1} + a_{n-2},\ n \geq 2$$

$$a_0 = 0,\ a_1 = 1.$$

Observação O algoritmo recursivo a seguir pode ser utilizado para obter qualquer elemento ou número de Fibonacci.

Algoritmo 2.1: Fibonacci.

% Entrada: uma função F e um número natural n.

% Saída: número de Fibonacci $F(n)$.

Início de função.

Se $n < 2$

então $F(n) := n$.

senão $F(n) := F(n-1) + F(n-2)$.

Fim de se-então-senão.

Fim de função.

Observação A virtude do algoritmo 2.1 repousa em sua elegância, já que expressa de forma sintética a relação matemática, porém, é extremamente ineficiente em termos computacionais. A principal fonte de ineficiência, no entanto, não está ligada unicamente à recursividade, mas à forma pela qual o algoritmo foi estruturado. Diversos valores são recomputados desnecessariamente, elevando em muito o tempo de computação.

Lema 2.1 Considere c_1 e c_2 como números reais. Suponha que

$$r^2 - c_1 r - c_2 = 0$$

tenha duas raízes distintas r_1 e r_2. Então, a sequência (a_n) é uma solução para a relação de recorrência

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

se, e somente se,

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

para $n = 0, 1, 2, \ldots$, em que α_1 e α_2 são constantes.

Teorema 2.2 A solução para a relação de recorrência de Fibonacci é a sequência

$$f_n = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n.$$

Demonstração A relação de recorrência de Fibonacci é dada por

$$f_n = f_{n-1} + f_{n-2}$$

com condição inicial, $f_0 = 0$ e $f_1 = 1$. Assim, tome $c_1 = 1$ e $c_2 = 1$. Segue-se que a equação $r^2 - r - 1 = 0$ possui as raízes

$$r_1 = \frac{1+\sqrt{5}}{2} \text{ e } r_2 = \frac{1-\sqrt{5}}{2},$$

as quais são distintas. Pelo lema 2.1, $f_n = \alpha_1 r_1^n + \alpha_2 r_2^n$. Para $n = 0$,

$$f_0 = \alpha_1 r_1^0 + \alpha_2 r_2^0 = \alpha_1 + \alpha_2 = 0$$

e, para $n = 1$,

$$f_1 = \alpha_1 r_1^1 + \alpha_2 r_2^1 = \alpha_1(\frac{1+\sqrt{5}}{2}) + \alpha_2(\frac{1-\sqrt{5}}{2}) = 1.$$

Resolvendo este sistema, obtemos $\alpha_1 = \frac{1}{\sqrt{5}}$ e $\alpha_2 = -\frac{1}{\sqrt{5}}$. Portanto

$$f_n = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n;$$

como queríamos demonstrar.

Observação O teorema 2.2 afirma que podemos encontrar o $n+1$-ésimo número de Fibonacci através da sequência de Fibonacci e não somente pela relação de recorrência de Fibonacci. Todavia, sua execução computacional esbarra no erro numérico para o cálculo do número irracional $\sqrt{5}$.

Exercícios para casa Fazer a Prova 3.

Próxima aula Paradigmas de projeto de algoritmos: algoritmo guloso.