Recorrências

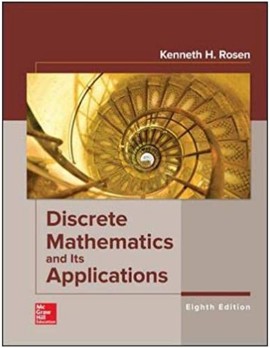
André Gustavo dos Santos[1]

[1] Departamento de Informática
Universidade Federal de Viçosa

INF230 - 2021/1

Conteúdo

Os slides seguintes são baseados nas seções 2.4.3 e 8.1 do livro texto da disciplina:



ROSEN, Kenneth H.
Discrete mathematics and its applications.
McGraw-Hill Education, 8th edition, 2018

Introdução

- Sequências são listas ordenadas de elementos
- São usadas em matemática discreta de muitas formas, por exemplo, para representar soluções de certos problemas de contagem
- Também são uma importante estrutura de dados
- Muitas vezes precisamos trabalhar com a soma dos elementos de uma sequência
- Os termos de muitas sequências podem ser definidos por uma fórmula
- Em alguns casos podem ser representados por uma relação de recorrência, que expressa cada elemento em termos de elementos anteriores

Sequências

- Uma sequência é uma estrutura discreta que representa uma lista ordenada
- Por exemplo, $1, 2, 3, 5, 8$ é uma sequência de 5 termos
- Já $1, 3, 9, 27, 81, \ldots, 3^n, \ldots$ é uma sequência infinita

Sequências - notação

- Usamos a_n para representar o n=ésimo elemento de uma sequência
- Usamos $\{a_n\}$ para representar a sequência

Sequência $\{a_n\}$ com $a_n = \frac{1}{n}$

- A lista de termos, começando com a_1, é dada por $a_1, a_2, a_3, a_4, \ldots$
- Ou seja, $\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$

Sequência $\{b_n\}$ com $b_n = 2 \cdot 5^n$

- A lista de termos, começando com b_0, é dada por $b_0, b_1, b_2, b_3, \ldots$
- Ou seja, $2, 10, 50, 250, \ldots$

Sequência $\{c_n\}$ com $c_n = -1 + 4n$

- A lista de termos, começando com c_0, é dada por $c_0, c_1, c_2, c_3, \ldots$
- Ou seja, $-1, 3, 7, 11, \ldots$

Recorrências

- Nos exemplos anteriores foram dadas fórmulas explícitas para os termos
- Outra forma de especificar uma sequência é informar um ou mais termos iniciais e uma regra para determinar os termos subsequentes a partir dos que os precedem
- Regras deste tipo são chamadas relações de recorrência

Relação de recorrência

Uma relação de recorrência para a sequência $\{a_n\}$ é uma equação que expressa a_n em termos de um ou mais termos anteriores da sequência (ou seja, $a_{n-1}, a_{n-2}, \ldots$) para todo inteiro $n \geq n_0$, sendo n_0 um inteiro não negativo.

Uma sequência é chamada solução de uma relação de recorrência se seus termos satisfazem a relação.

Sequência $\{a_n\}$ com $a_n = a_{n-1} + 3$ e $a_0 = 2$. Quanto valem a_1, a_2, a_3?

- $a_1 = a_0 + 3 = 2 + 3 = 5$
- $a_2 = a_1 + 3 = 5 + 3 = 8$
- $a_3 = a_2 + 3 = 8 + 3 = 11$

UFV

Recorrências - exemplos

Sequência $\{b_n\}$ com $b_n = 5 \cdot b_{n-1}$ e $b_0 = 2$. Quanto valem b_1, b_2, b_3?

- $b_1 = 5 \cdot a_0 = 5 \cdot 2 = 10$
- $b_2 = 5 \cdot a_1 = 5 \cdot 10 = 50$
- $b_3 = 5 \cdot a_2 = 5 \cdot 50 = 250$

Sequência $\{c_n\}$ com $c_n = 4 + c_{n-1}$ e $c_0 = -1$. Quanto valem c_1, c_2, c_3?

- $c_1 = 4 + c_0 = 4 + (-1) = 3$
- $c_2 = 4 + c_1 = 4 + 3 = 7$
- $c_2 = 4 + c_2 = 4 + 7 = 11$

Note que estas relações de recorrência definem $\{b_n\}$ e $\{c_n\}$ dadas como exemplo anteriormente.

Recorrências - exemplos

Sequência $\{f_n\}$ com $f_n = f_{n-1} + f_{n-2}$ e $f_1 = 1$ e $f_2 = 2$. Quanto valem f_3, f_4, f_5?

- $f_3 = f_2 + f_1 = 2 + 1 = 3$
- $f_4 = f_3 + f_2 = 3 + 2 = 5$
- $f_5 = f_4 + f_3 = 5 + 3 = 8$

Note que os 5 primeiros termos desta sequência são os termos da primeira sequência dada como exemplo nestes slides

Recorrências

- Nos exemplos anteriores vimos que uma relação de recorrência define recursivamente uma sequência
- As condições iniciais de uma sequência definida recursivamente especificam os termos que precedem o primeiro termo em que a recorrência surte efeito
- Por exemplo, $f_1 = 1$ e $f_2 = 2$, $c_3 = -1$, $b_3 = 2$, $a_0 = 2$ nos exemplos anteriores

- Quando resolvemos uma relação de recorrência com as condições iniciais, achamos uma fórmula fechada para os termos da sequência
- Ou seja, uma fórmula explícita para os termos, que não usa termos anteriores
- Técnicas de solução de relações de recorrência são vistas a seguir

Métodos de resolução - *forward substitution*

- Nesta técnica de substituição para frente, começamos com a condição inicial e calculamos os termos subsequentes usando a relação de recorrência
- Continuamos as substituições para frente até ficar clara a fórmula fechada para a_n

Resolva a relação de recorrência $a_n = a_{n-1} + 3$ para $n = 1, 2, \ldots$ com $a_0 = 2$

Temos que

- $a_0 = 2$
- $a_1 = a_0 + 3 = 2 + 3$
- $a_2 = a_1 + 3 = 2 + 3 + 3 = 2 + 2 \cdot 3$
- $a_3 = a_2 + 3 = 2 + 2 \cdot 3 + 3 = 2 + 3 \cdot 3$
- $a_4 = a_3 + 3 = 2 + 3 \cdot 3 + 3 = 2 + 4 \cdot 3$
- $\ldots$
- $a_n = 2 + n \cdot 3$

- Note que no passo final a fórmula foi simplesmente “adivinhada” a partir do padrão dos termos iniciais; ainda deve ser provada por indução

Métodos de resolução - *backward substitution*

- Na técnica de substituição para trás, iniciamos com a_n e expressamos seu valor em função dos termos anteriores até ser expresso em termos da condição inicial

Resolva a relação de recorrência $a_n = a_{n-1} + 3$ para $n = 1, 2, \ldots$ com $a_0 = 2$

Temos que

- $$\begin{aligned} a_n &= a_{n-1} + 3 \\ &= (a_{n-2} + 3) + 3 = a_{n-2} + 2 \cdot 3 \\ &= (a_{n-3} + 3) + 2 \cdot 3 = a_{n-3} + 3 \cdot 3 \\ &= (a_{n-4} + 3) + 3 \cdot 3 = a_{n-4} + 4 \cdot 3 \\ &= \ldots \\ &= a_1 + (n-1) \cdot 3 = (a_0 + 3) + (n-1) \cdot 3 \\ &= 2 + n \cdot 3 \end{aligned}$$

UFV

Exemplo

Juros compostos

- Uma pessoa aplica \$10.000 em uma conta poupança que rende 11% ao ano.
- Quanto ela terá na conta depois de 30 anos?
- Solução:
  - Seja P_n o valor que ela terá depois de n anos
  - A sequência $\{P_n\}$ satisfaz a recorrência $P_n = P_{n-1} + 0.11P_{n-1} = 1.11P_{n-1}$
  - A condição inicial é $P_0 = 10000$
  - Resolvendo com o método de substituição para frente temos que:
    $P_1 = 1.11P_0$
    $P_2 = 1.11P_1 = (1.11^2)P_0$
    $P_3 = 1.11P_2 = (1.11^3)P_0$
    $\ldots$
    $P_n = 1.11P_{n-1} = (1.11^n)P_0$
  - Depois de 30 anos a pessoa terá $P_{30} = 1.11^{30} \cdot 10000 = \$228.922,97$

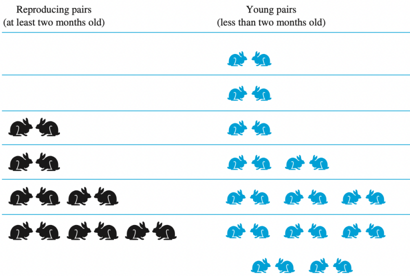
Aplicações de recorrência

- Relações de recorrência são muito importantes no estudo de algoritmos
- Programação dinâmica
  - Importante paradigma de desenvolvimento de algoritmos
  - Algoritmos que usam esta técnica dividem um problema em subproblemas sobrepostos
  - A solução do problema é obtida das soluções dos subproblemas por uma relação de recorrência
- Divisão-e-Conquista
  - Outro importante paradigma de desenvolvimento de algoritmos
  - Algoritmos com esta técnica dividem um problema em subproblemas não sobrepostos
  - Os subproblemas são divididos até que possam ser resolvidos diretamente
  - A análise de complexidade desses algoritmos é feita por relações de recorrência
- Relações de recorrência são úteis na solução de problemas de contagem
  - Alguns podem ser modelados por relações de recorrência
  - Alguns podem ser resolvidos utilizando séries de potências, as funções geradoras
  - Os coeficientes das potências de x representam os termos da sequência
  - Funções geradoras podem ser usados também para resolver relações de recorrência

Modelos com relações de recorrência

Contagem de coelhos – proposto por Leonardo Pisano, *Liber abaci*, 1202

Um jovem casal de coelhos é colocado numa ilha. Um casal de coelhos não procria até que eles tenham 2 meses de idade. Depois de completar 2 meses, cada casal de coelhos procria outro casal a cada mês, como mostrado na figura. Encontre uma relação de recorrência para o número de casais de coelhos na ilha depois de *n* meses, supondo que nenhum coelho morra.



Month	Reproducing pairs	Young pairs	Total pairs
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

FIGURE 1 **Rabbits on an island.**

Modelos com relações de recorrência

Contagem de coelhos – proposto por Leonardo Pisano, *Liber abaci*, 1202

Um jovem casal de coelhos é colocado numa ilha. Um casal de coelhos não procria até que eles tenham 2 meses de idade. Depois de completar 2 meses, cada casal de coelhos procria outro casal a cada mês, como mostrado na figura. Encontre uma relação de recorrência para o número de casais de coelhos na ilha depois de *n* meses, supondo que nenhum coelho morra.

- Seja f_n o número de casais de coelhos depois de n meses
- Ao final do primeiro mês há apenas o casal levado pra ilha, então $f_1 = 1$
- Este casal não procria durante o segundo mês, então $f_2 = 1$ também
- Depois de n meses temos todos os casais do mês anterior, f_{n-1}, e os casais recém-nascidos, f_{n-2}, porque cada um vem de um casal com pelo menos 2 meses de idade: $f_n = f_{n-1} + f_{n-2}$
- A sequência $\{f_n\}$ satisfaz a recorrência $f_n = f_{n-1} + f_{n-2}$ com condições iniciais $f_1 = 1$, $f_2 = 1$
- Ela é chamada sequência de Fibonacci, em homenagem a Leonardo Fibonacci, conhecido por Leonardo Pisano, de Pisa, que propôs o problema acima
- Aparece em diversos situações na matemática e na computação e pode ser observada em muitos contextos na natureza.

UFV

Modelos com relações de recorrência

Torres de Hanoi – Édouard Lucas, 1883

Uma base contém três pinos e em um deles estão dispostos n discos uns sobre os outros, em ordem de tamanho. Como passar todos os discos desse pino para um outro (usando um dos pinos como auxiliar), de maneira que um disco nunca fique em cima de outro menor que ele?

- Uma forma de passar todos os n discos do pino 1 para o pino 2 é:
  - Mover os $n-1$ menores do pino 1 para o 3 (usando o pino 2 como auxiliar)
  - Mover o maior (o n-ésimo) para o pino 2
  - Mover os $n-1$ menores do pino 3 para o 2 (usando o pino 1 como auxiliar)
- Quantos movimentos são feitos?

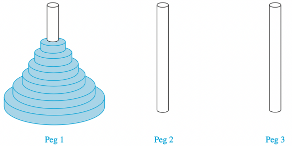


FIGURE 2 The initial position in the Tower of Hanoi.

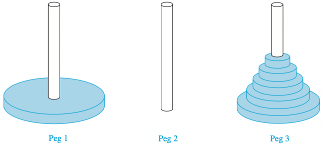


FIGURE 3 An intermediate position in the Tower of Hanoi.

UFV

Modelos com relações de recorrência

Torres de Hanoi – Édouard Lucas, 1883

Uma base contém três pinos e em um deles estão dispostos n discos uns sobre os outros, em ordem de tamanho. Como passar todos os discos desse pino para um outro (usando um dos pinos como auxiliar), de maneira que um disco nunca fique em cima de outro menor que ele?

- Uma forma de passar todos os n discos do pino 1 para o pino 2 é:
  - Mover os $n-1$ menores do pino 1 para o 3 (usando o pino 2 como auxiliar)
  - Mover o maior (o n-ésimo) para o pino 2
  - Mover os $n-1$ menores do pino 3 para o 2 (usando o pino 1 como auxiliar)
- Quantos movimentos são feitos?
  - Seja H_n o número de movimentos para reposicionar n discos de um pino a outro
  - No algoritmo acima, $H_n = H_{n-1} + 1 + H_{n-1} = 2H_{n-1} + 1$
  - O número de movimentos é a solução da recorrência $H_n = 2H_{n-1} + 1$ com $H_1 = 1$
  - Pelo método de substituição para trás temos que:

$$
\begin{aligned}
H_n &= 2H_{n-1} + 1 \\
&= 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\
&= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 4 + 2 + 1 \\
&= 2^2(2H_{n-4} + 1) + 4 + 2 + 1 = 2^4 H_{n-4} + 8 + 4 + 2 + 1 \\
&\ldots \\
&= 2^{n-1} H_1 + 2^{n-2} + 2^{n-3} + \cdots + 8 + 4 + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 8 + 4 + 2 + 1 \\
&= 2^n - 1.
\end{aligned}
$$

Modelos com relações de recorrência

Quantos strings de n bits sem 0's consecutivos existem? Quantos de tamanho 5?

- Seja a_n o número de strings de n bits sem 0's consecutivos
- Note que $a_1 = 2$, strings `0` e `1`, e que $a_2 = 3$, strings `01`, `10` e `11`
- Para $n \geq 3$, contaremos separadamente os que terminam com `1` e os com `0`
- Os que terminam com `1` sem 0's consecutivos são todos os de tamanho $n-1$ bits sem 0's consecutivos com um `1` acrescentado no final. Portanto, a_{n-1}.
- Os que terminam com `0` sem 0's consecutivos devem ter um `1` na penúltima posição, senão haveria um `00` no final. Então são todos de tamanho $n-2$ bits sem 0's consecutivos com `10` acrescentado no final. Portanto, a_{n-2}.
- Somando-se estes valores temos que $a_n = a_{n-1} + a_{n-2}$
- A solução então é dada pela sequência $\{a_n\}$ definida pela relação de recorrência $a_n = a_{n-1} + a_{n-2}$ com condições inicias $a_1 = 2$ e $a_3 = 3$
- Note que é mesma recorrência da sequência de Fibonacci
- Pela definição de $\{a_n\}$ obtemos $a_5 = 13$, existem 13 de tamanho 5

Modelos com relações de recorrência

Enumeração de códigos

Certo sistema computacional considera um string de dígitos decimais como um código válido se contém um número par de dígitos 0. Por exemplo, 1230407869 é válido, enquanto 120987045608 não. Encontre uma relação de recorrência para o número de códigos válidos de n dígitos.

- Seja a_n o número de códigos válidos de n dígitos
- $a_1 = 9$, pois há 10 códigos de 1 dígito, mas um deles, o 0, não é válido
- Para encontrar a recorrência, devemos considerar como um código válido de n dígitos pode ser formado a partir de códigos de $n-1$ dígitos. Há duas formas:
  - Acrescentando um dígito, exceto 0, depois de um código válido de $n-1$ dígitos
  - Acrescentando um 0 depois de um código inválido de $n-1$ dígitos
- A primeira forma está correta porque não muda o número de 0's
- A segunda poque um código inválido tem número ímpar de 0's; com mais um 0, fica válido
- Há 9 formas de se acrescentar um dígito em código válido e há a_{n-1} códigos válidos de $n-1$ dígitos, então há $9 \cdot a_{n-1}$ códigos de n dígitos formados da primeira forma
- Há 1 forma de acrescentar 0 em código inválido e há $10^{n-1} - a_{n-1}$ códigos inválidos de $n-1$ dígitos (pois há 10^{n-1} no total e a_{n-1} são válidos)
- Concluindo, $a_n = 9 \cdot a_{n-1} + 10\hat{}n - 1 - a_{n-1} = 8 \cdot a_{n-1} + 10^{n-1}$.

Modelos com relações de recorrência

Parênteses em expressões

Encontre uma relação de recorrência para C_n, o número de maneiras de colocar parênteses num produto de $n+1$ números, $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$, para especificar a ordem de multiplicação.

Por exemplo, $C_3 = 5$, porque há 5 formas de colocar parênteses em $x_0 \cdot x_1 \cdot x_2 \cdot x_3$:
$((x_0 \cdot x_1) \cdot x_2) \cdot x_3$; $(x_0 \cdot (x_1 \cdot x_2) \cdot x_3)$; $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$; $x_0 \cdot ((x_1 \cdot x_2) \cdot x_3)$; $x_0 \cdot (x_1 \cdot (x_2 \cdot x_3))$

- Ao colocar parênteses, um "$\cdot$" fica fora dos parênteses, a última multiplicação realizada
- Este "$\cdot$" entre dois números x_k e x_{k+1} deixa $k+1$ números de um lado e $n-k$ do outro
- Pela regra do produto, há $C_k C_{n-k-1}$ maneiras com última multiplicação entre x_k e x_{k+1}
- Pela regra da soma, considerando todas as posições da última multiplicação, temos então:

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-k-1}$$

com condições iniciais $C_0 = 1$ e $C_1 = 1$.

- A sequência $\{C_n\}$ é a sequência dos números de Catalan
- Eles aparecem em vários outros problemas de contagem

Paradigma de projeto de algoritmos - Divisão-e-Conquista

Quantas comparações, no pior caso, faz uma busca binária numa lista de $n = 2^k$ elementos?

- Seja a_n o número de comparações para n elementos, no pior caso
- Note que o enunciado assume que n é uma potência de 2
- Para n elementos a busca binária verifica o valor do elementos "do meio" ($n/2$ ou $n/2+1$) e continua, no pior, caso, com exatamente a metade dos elementos
- Então $a_n = 1 + a_{n/2}$ com condição inicial $a_1 = 1$
- Pelo método de substituição para trás:
  - $$\begin{aligned} a_n &= 1 + a_{n/2} \\ &= 1 + (1 + a_{n/2^2}) = 2 + a_{n/2^2} \\ &= 2 + (1 + a_{n/2^3}) = 3 + a_{n/2^3} \\ &= 3 + (1 + a_{n/2^3}) = 4 + a_{n/2^4} \\ &\ldots \\ &= i + a_{n/2^i} \\ &= \log_2 n + a_1 \qquad \textit{(note que } n/2^i = 1 \textit{ quando } n = 2^i\textit{, ou seja, } i = \log_2 n \\ &= \log_2 n + 1. \end{aligned}$$

UFV

Paradigma de projeto de algoritmos - Programação Dinâmica

Encontrar a sequência de tarefas que traz o maior retorno total

- Cada tarefa j tem hora de início s_j e de fim t_j definidas
- Cada tarefa j tem um retorno w_j
- Só pode realizar tarefas compatíveis, isto é, que não se sobrepõem no tempo
- Quais tarefas realizar para obter o maior retorno possível?

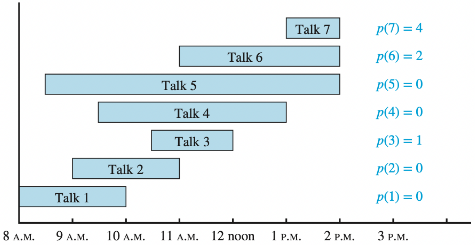


FIGURE 5 **A schedule of lectures with the values of $p(n)$ shown.**

- Seja T_n o máximo possível com as tarefas $1, 2, \ldots, n$
- $T_n = \max(w_n + T_{p(n)}, T_{n-1})$ sendo $p(n)$ a última tarefa compatível com n e $T_0 = 0$.