Lista 21

21.2. Prove a Equação (22).

21.3. Prove as igualdades a seguir por indução. Em cada caso, n é um inteiro positivo.

a. $1 + 4 + 7 + \cdots + (3n - 2) = \frac{n(3n-1)}{2}$.

b. $1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}$.

c. $9 + 9 \times 10 + 9 \times 100 + \cdots + 9 \times 10^{n-1} = 10^n - 1$.

d. $\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \cdots + \frac{1}{n(n+1)} = 1 - \frac{1}{n+1}$.

21.4. Prove as igualdades a seguir por indução. Em cada caso, n é um inteiro positivo.

a. $2^n \leq 2^{n+1} - 2^{n-1} - 1$

b. $(1 - \frac{1}{2})(1 - \frac{1}{4})(1 - \frac{1}{8}) \cdots (1 - \frac{1}{2^n}) \geq \frac{1}{4} + \frac{1}{2^{n+1}}$.

c. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{2^n} \geq 1 + \frac{n}{2}$.

21.5. Um grupo de pessoas está em uma fila para comprar entradas para um cinema. A primeira pessoa na fila é uma mulher e a última é um homem. Aplique a prova por indução para mostrar que, em algum ponto da fila, uma mulher está diretamente na frente de um homem.

21.6. A *Torre de Hanói* é um jogo que consiste em um tabuleiro com três espigões e uma coleção de n discos de tamanhos (raios) diferentes. Os discos têm orifícios perfurados em seus centros, de modo a poderem adaptar-se aos espigões no tabuleiro. Inicialmente, todos os discos estão no primeiro espigão, dispostos por tamanho (do maior, na base, para o menor, no topo).

O objetivo é transferir todos os discos para outro espigão com o menor número possível de movimentos. Cada movimento consiste em tirar o disco de cima de um dos espigões e colocá-lo em outro espigão, com a condição de não se colocar um disco maior em cima de um disco menor. A figura mostra como resolver o problema da Torre de Hanói em três movimentos quando $n = 2$.

Prove: Para todo inteiro positivo n, o jogo da Torre de Hanói (com n discos) pode ser resolvido com $2^n - 1$ movimentos.

21.7. Sejam $A_1, A_2, \ldots, A_n$ conjuntos (com $n \geq 2$). Suponha que, para dois conjuntos quaisquer A_i e A_j, ou $A_i \subseteq A_j$ ou $A_j \subseteq A_i$.

Prove, por indução, que um desses n conjuntos é um subconjunto de todos eles.

A estreita relação entre definição por recorrência e prova por indução.

21.8. Uma palavra pode ser usada em sua própria definição? Em geral, a resposta é não. Todavia, na Definição 20.12, definimos os números de Fibonacci como a sequência $F_0, F_1, F_2, \ldots$, fazendo $F_0 = 1$, $F_1 = 1$ e, para $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. Note que definimos os números de Fibonacci em termos deles próprios! Isso funciona porque definimos F_n em termos de números de Fibonacci previamente definidos. Esse tipo de definição é chamado definição *por recorrência*.

As definições por recorrência guardam forte semelhança com as provas por indução. Elas são tipicamente alguns casos básicos, e o resto da definição se reporta a casos menores (à semelhança com a etapa indutiva de uma prova por indução).

A indução é a técnica de prova de escolha para provar afirmações sobre conceitos definidos por recorrência.

As seguintes sequências de números são definidas por recorrência. Responda às questões formuladas.

a. Seja $a_0 = 1$ e, para $n > 0$, seja $a_n = 2a_{n-1} + 1$. Os primeiros termos da sequência, a_0, a_1, a_2, a_3 ... são 1, 3, 7, 15, ...
Quais são os próximos três termos?
Prove: $a_n = 2_{n+1} - 1$.

b. Seja $b_0 = 1$ e, para $n > 0$, seja $b_n = 3b_{n-1} - 1$.
Quais são os cinco primeiros termos da sequência b_0, b_1, b_2, ... ?
Prove: $b_n = \frac{3^n+1}{2}$.

c. Seja $c_0 = 3$ e, para $n > 0$, $c_n = c_{n-1} + n$.
Quais são os cinco primeiros termos da sequência c_0, c_1, c_2, ...?
Prove: $c_n = \frac{n^2+n+6}{2}$

d. Seja $d_0 = 2$, $d_1 = 5$ e, para $n > 1$, seja $d_n = 5d_{n-1} - 6d_{n-2}$.
Por que damos duas definições-base?
Quais são os cinco primeiros termos da sequência d_0, d_1, d_2, ...?
Prove que $d_n = 2_n + 3_n$.

e. Seja $e_0 = 1$, $e_1 = 4$ e, para $n > 1$, $e_n = 4(e_{n-1} - e_{n-2})$.
Quais são os cinco primeiros termos da sequência e_0, e_1, e_2, ...?
Prove que $e_n = (n + 1)2^n$.

f. Denote por F_n o n-ésimo número de Fibonacci. Prove:

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

21.9. Um mastro tem n pés de altura. Nesse mastro, exibimos bandeiras dos seguintes tipos: bandeiras vermelhas com 1 pé de altura, bandeiras azuis com 2 pés de altura e bandeiras verdes com 2 pés de altura. A soma das alturas das bandeiras é exatamente n pés.

Prove que há $\frac{2}{3}2^n + \frac{1}{3}(-1)^n$ maneiras de dispor as bandeiras.

21.10. Prove que todo inteiro positivo pode ser expresso como a soma de números de Fibonacci distintos.

Por exemplo, $20 = 2 + 5 + 13$, onde 2, 5, 13 são, obviamente, números de Fibonacci. Embora possamos escrever $20 = 2 + 5 + 5 + 8$, isso não ilustra o resultado, porque utilizamos 5 duas vezes.

21.11. Consideremos o seguinte programa de computador:

```
function findMax(array, first, last) {
  if (first == last) return array[first];
  mid = first + (last-first)/2;

  a = findMax (array, first, mid);
  b = findMax (array, mid+1, last);
  if (a<b) return b;
  return a;
}
```

Aqui, `array` é uma fileira de inteiros. Todas as outras variáveis são inteiros. Supomos que `first` e `last` estejam entre 1 e o número de elementos da fileira e que `first ≤ last`.

O objetivo deste programa é achar o maior valor na fileira entre dois índices; isto é, o programa deve dar o maior valor de `array[first], array[first + 1], ..., array[last]`.

Seu trabalho: provar que esse programa desempenha sua tarefa.

[*Nota técnica*: Se `last-first` é ímpar, então `(last-first)/2` é arredondado para o inteiro inferior mais próximo. Por exemplo, se `first` é 7 e `last` é 20, então `(last-first)/2 = 6`.]

21.12. Considere o seguinte programa de computador:

```
function lookUp (array, first, last, key) {
   mid = first + (last-first)/2;
   if (array [mid] == key) return mid;
   if (array [mid] > key) return lookUp (array, first, mid-1,
   key);
   return lookUp (array, mid+1, last, key);
}
```

Aqui, `array` é uma fileira de inteiros; todas as outras variáveis representam inteiros. Os valores armazenados em array são escolhidos; isto é, sabemos que

```
array [1] < array [2] < array [3] < . . . .
```

Sabemos também que $1 \leq$ `first` $\leq$ `last` e que há um índice j entre `first` e `last` para o qual o array [j] é igual a `key`.

Prove que esse programa acha aquele índice j.

21.13. Procure provar, utilizando a indução forte ou a indução-padrão, que um polígono triangulado tem ao menos um triângulo exterior.

O que acontece de errado quando procuramos elaborar nossa própria prova?

O teorema mais difícil ("... tem ao menos dois triângulos exteriores") é mais fácil de provar do que o teorema mais fácil ("... tem ao menos um triângulo exterior"). Esse fenômeno é conhecido como carga de indução.

21.14. Prove o Teorema 21.9.

21.15. Utilizando a indução forte, prove que todo número natural pode ser expresso como a soma de potências distintas de 2. Por exemplo, $2^1 = 2^4 + 2^2 + 2^0$.

21.16. Mostramos como provar o princípio da indução matemática (Teorema 21.2) utilizando o Princípio da Boa Ordenação. Faça agora o oposto: aplique a indução para provar o Princípio da Boa Ordenação (Afirmação 20.6).