MC 358 - Lista de exercícios 3

1. O máximo divisor comum de dois inteiros a e b, denotado por mdc(a, b), é definido como o maior inteiro que divide ambos a e b. Sejam a, b, d ∈ Z. Teorema (Identidade de Bézout): Se mdc(a, b) = d, então existem u, v ∈ Z tais que d = u · a + v · b.

(a) Escreva a recíproca do Teorema acima. Prove a veracidade da proposição que você escreveu ou apresente um contraexemplo.
(b) Para o caso em que d = 1, prove a veracidade da proposição que você escreveu no item anterior ou apresente um contraexemplo.

Definindo a proposição original como P(a,b,d,u,v):
$P(a,b,d,u,v) = mdc(a,b) = d \rightarrow \exists\, u,v \in \mathbb{Z}(d = u.a + v.b)$

Definindo a recíproca como Q(a,b,d,u,v):
$Q(a,b,d,u,v) = \exists\, u,v \in \mathbb{Z}(d = u.a + v.b) \rightarrow mdc(a,b) = d$

A existência de dois inteiros que satisfaçam a equação para d, não implica necessariamente que d seja o mdc de a e b. Se a e b são primos entre si, ou seja, a e b são divisíveis apenas por 1 e por eles mesmos, nesse caso o mdc(a,b) = 1.
Tomando dois exemplos:
a = 2 e b = 3, mdc(2,3) = 1, 2u + 3v = 1 (I)
a = 5 e b = 7, mdc(5,7) = 1, 5u + 7v = 1 (II)

2u + 3v = 5u + 7v
-3u = 4v
u = - 4v/3

substituindo no exemplo(I):
2(-4v/3) + 3v = 1
-8v/3 + 3v = 1
(-8v + 9v)/3 = 1
1/3 = 1
Ao tentarmos resolver o sistema, sempre chegaremos a uma indefinição, pois não existe solução inteira que satisfaça as equações.

2. Demonstre as seguintes afirmações sobre divisibilidade:
(a) Se mdc(a, n) = 1 e n | (a · b), então n | b.

Se o mdc(a,n) = 1, então a e n são primos entre si, ou seja, ambos são primos e possuem 1 como maior divisor comum.

Pelo Teorema de Bézout, podemos escrever:

$\exists\, x,y \in \mathbb{Z}$, ax + ny = 1

Multiplicando ambos os lados por b:

abx + nby = b
Se n | (a.b), podemos escrever:
∃ z ∈ ℤ, nz = ab
Substituindo ab na equação anterior:
nzx + nby = b
n(zx + by) = b
Como x, y e z são inteiros, o termo (zx + by) é inteiro. Portanto b é múltiplo de n e n | b.

(b) mdc(a, b) = 1, se, e somente se, mdc(a, b^n) = 1 para todo natural n ≥ 1.

Definindo a proposição T(a,b,n):
T(a,b,n) = "mdc(a,b) = 1 ⇔mdc(a, b^n) = 1, para todo natural n >= 1"
Pelas Leis do Operador de Equivalência, podemos dividir a proposição na conjunção de duas outras proposições:
P(a,b,n) = "mdc(a,b) = 1 → mdc(a,b^n) = 1, para todo natural n >= 1" e
Q(a,b,n) = "mdc(a,b^n) = 1 → mdc(a,b) = 1, para todo natural n >= 1"

Pelo Teorema de Bézout podemos afirmar:
mdc(a,b) = 1 então ∃ x,y ∈ ℤ, ax + by = 1
mdc(a,b^n) = 1 então ∃ x',y' ∈ ℤ, ax' + (b^n)y' = 1

Pela Lei do Silogismo Hipotético:
mdc(a,b) = 1 →mdc(a, b^n) = 1 e mdc(a,b^n) = 1 →ax'+(b^n)y' = 1
então mdc(a,b) = 1 → ax' + (b^n)y' = 1

mdc(a,b^n) = 1 →mdc(a, b) = 1 e mdc(a,b) = 1 →ax+by = 1
então mdc(a,b^n) = 1 → ax + by = 1

(c) Se mdc(a, n) = 1 e n | (a^k · b), para algum inteiro k ≥ 1, então n | b.

Pelo Teorema de Bézout, podemos escrever:
Se o mdc(a,n) = 1, então ∃ x,y ∈ ℤ, ax + ny = 1
Multiplicando ambos os lados por b:
abx + nby = b
abx = b - nby
a = (b - nby)/bx
Elevando ambos os lados a k:
a^k = [(b-nby)/bx]^k

Se n | (a^k . b), podemos escrever:
∃ z ∈ ℤ, nz = a^k . b
a^k = nz/b

Substituindo:
nz/b = [(b-nby)/bx]^k
n = [(bx(nz/b)^1/k) - b]/by

Os *números de Fibonacci* $F_0, F_1, F_2, \ldots$ são definidos pelas seguintes regras:

$$F_0 = 0,\ F_1 = 1 \text{ e } F_n = F_{n-1} + F_{n-2} \text{ para } n \geq 2.$$

---

3. **Prove por indução completa** que $F_n < \left(\frac{13}{8}\right)^n$ para todo inteiro $n \geq 0$.

Definindo a proposição P(n):
P(n) = Fn < (13/8)^n para todo inteiro n >= 0

Caso base:

Queremos provar P(0) e P(1):
P(0) = Fn < $(13/8)^0$
Fn < 1 , como F0 = 0 , 0 < 1

P(1) = Fn < $(13/8)^1$
Fn < 13/8, como F1 = 1, 1 < 13/8 CQD(Base)

Hipótese indutiva:

Supomos que existe um k >= 2 tal que vale, P(0) e P(1) e … e P(k)

Passo indutivo:

Queremos provar P(k+1) = Fk+1 < (13/8)^k+1

Usando a definição da sequência de Fibonacci:
Fk = Fk-1 + Fk-2 , para k >= 2
Fk+1 = Fk + Fk-1

Como a hipótese vale para k >= 2(vale para k=2 e para k-1=1), podemos escrever pela hipótese de indução:
Fk < (13/8)^k e Fk-1 < (13/8)^k-1
Somando as desigualdades obtemos:
Fk + Fk-1 < (13/8)^k + (13/8)^k-1 , sabemos que Fk + Fk-1 = Fk+1
Fk+1 < (13/8)^k + (13/8)^k-1

Temos que (13/8)^k + (13/8)^k-1 pode ser escrito como:
[(13/8)^k+1 (13/8)^-1] + [(13/8)^k+1 (13/8)^-2] , colocando o termo que se repete em evidência:
(13/8)^k+1 [(13/8)^-1 + (13/8)^-2] = (13/8)^k+1 (8/13 + 64/169)

Como a fração (8/13 + 64/169) resulta em um valor menor que 1, o termo (13/8)^k+1 multiplicado por essa fração, sempre resultará em um valor menor que o termo original. Portanto:

(13/8)^k+1 (8/13 + 64/169) < (13/8)^k+1
(13/8)^k + (13/8)^k-1 < (13/8)^k+1

Como Fk+1 < (13/8)^k + (13/8)^k-1. Então:
Fk+1 < (13/8)^k+1 CQD(Passo).

4. Prove por indução completa que para quaisquer números naturais n e m, vale que Fm · Fn + Fm+1 · Fn+1 = Fm+n+1. Dica: tome um m arbitrário e prove por indução em n.

Definindo a proposição P(m,n) = Fm . Fn + Fm+1 . Fn+1 = Fm+n+1, para todos naturais n e m.
Tomando um m arbitrário e provando por indução em n.

Caso base:
Queremos provar P(m,0) e P(m,1):
P(m,0) = Fm . F0 + Fm+1 . F1 = Fm . 0 + Fm+1 . 1 = Fm+1 = Fm+0+1
P(m,1) = Fm . F1 + Fm+1 . F2 = Fm + Fm+1 . (F1 + F0) = Fm + Fm+1 . 1 = Fm + Fm+1 = Fm+1+1 = Fm+1 + Fm CQD(Base)

Hipótese indutiva:
Supomos que existe um k >= 2 tal que vale, P(m,0) e P(m,1) e … e P(m,k)
Passo indutivo:
Queremos provar P(m,k+1) = Fm . Fk+1 + Fm+1 . Fk+2 = Fm+k+2

Utilizando a definição da sequência de Fibonacci:
Fk = Fk-1 + Fk-2 , para k >= 2
Fk+1 = Fk + Fk-1
Fk+2 = Fk+1 + Fk

Pela hipótese de indução, a proposição vale para k e k-1, portanto podemos escrever:
Fm . (Fk + Fk-1) + Fm+1 . (Fk+1 + Fk) =
Fm . Fk + Fm . Fk-1 + Fm+1 . Fk+1 + Fm+1 . Fk =
Fm+k+1 + Fm . Fk-1 + Fm+1 . Fk =
Fm+k+1 + Fm+k = Fm+k+2 CQD (Passo)

5. Use indução completa para mostrar que qualquer inteiro positivo pode ser escrito como uma soma de potências de 2 distintas.

A proposição descrita define a representação binária de decimais inteiros.
Definindo a proposição:

P(n) = ∀ n ∈ $\mathbb{Z}$+ ($\sum_{i=0}^{j}$ ai . 2^i = n, para j inteiro positivo e ai assumindo valores 0 ou 1 para cada potência i de 2).

Caso base:

Queremos provar P(1):

P(1) = $\sum_{i=0}^{0}$ ai . 2^i = 1. 2^0 = 1 CQD(Base)

Hipótese indutiva:

Supomos que para um k inteiro positivo vale, P(1) e P(2) e … e P(k)

Passo indutivo:

Queremos provar P(k+1) = $\sum_{i=0}^{j}$ ai . 2^i = k+1

Temos dois casos:

Caso 1: k+1 é um inteiro positivo ímpar

Nesse caso, k é par. Pela hipótese de indução, k pode ser escrito como a soma de potências de dois distintas. Porém, se k é par, então k tem uma representação binária com o dígito menos significativo igual a 0. Portanto, P(1) não faz parte da soma, então P(k+1) = P(k) + P(1) é uma representação com somas de potências de 2 distintas.

Caso 2: k+1 é um inteiro positivo par

Nesse caso, (k+1)/2 é um inteiro positivo entre 1 e k. Portanto, pela hipótese de indução, (k+1)/2 pode ser escrito como uma soma de potências de dois distintas. Assim, k+1 pode ser escrito como P(k+1) = 2 . P((k+1)/2), formado por potências de dois distintas. CQD(Passo)

6. Suponha que há uma pilha formada por n ≥ 1 blocos e que você deseja transformá-la em n pilhas contendo um único bloco. Para atingir esse objetivo, você pretende empregar o seguinte procedimento sucessivas vezes: escolher uma pilha com pelo menos dois blocos e dividi-la em outras duas pilhas menores. Cada vez que você divide uma pilha, você calcula r·s, onde r e s são as quantidades de blocos das duas pilhas resultantes da divisão. Prove por indução completa que independentemente das suas escolhas, a soma dos produtos calculados em cada etapa será igual a n(n − 1)/2.

Definindo a proposição P(n) que representa a soma dos produtos r.s em cada etapa da divisão de blocos:

P(n) = n(n-1)/2

Caso base:

Queremos provar que a proposição é válida para uma pilha de 1 bloco:

Caso a pilha tenha 1 bloco, não ocorrem divisões, portanto r ou s será igual a zero, assim r.s = 0 = 1(1-1)/2 = 0. CQD(Base)

Hipótese indutiva:

Supomos que para um k >= 1 vale, P(1) e P(2) e … e P(k) = k(k-1)/2

Passo indutivo:

Queremos provar P(k+1) = k+1(k+1-1)/2 = k(k+1)/2

Se temos uma pilha com k+1 blocos, ao realizar a primeira divisão, obtemos duas novas pilhas com r e s blocos, portanto r+s = k+1 e r.s é adicionado a lista de produtos a serem somados.

Após isso, temos duas novas pilhas que serão divididas na sequência até que cada bloco esteja em sua própria pilha. Sabemos que todas as pilhas seguintes, serão subdivisões de uma pilha de k+1 blocos. Podemos então dizer que k+1 será dividido por um x natural tal que x > 1: $(k+1)/x = k/x + 1/x$. Como k é sempre um número natural e $k/x < k$ e $1/x < 1$, então $(k+1)/x < k$.

Pela hipótese de indução, conseguimos resolver o problema para qualquer pilha de 1 até k blocos. Assim, podemos resolver para qualquer subdivisão de (k+1)/x. Portanto, podemos resolver para as pilhas de r e s blocos, a soma dos produtos das divisões será então:

$r(r-1)/2 + s(s-1)/2 + rs$
$= (r^2 - r + s^2 - s + 2rs)/2$
$= [(r+s)^2 - (r+s)]/2$, substituindo r+s = k+1
$= [(k+1)^2 - (k+1)]/2$
$= (k^2 + 2k + 1 - k - 1)/2$
$= (k^2 + k)/2$
$= k(k+1)/2$ CQD(Passo)

8. Sabendo que o algoritmo recursivo `ABRACADABRA` tem como resultado o produto de todos os elementos de A (fazendo $f = n - 1$ na primeira chamada), **prove a sua corretude utilizando indução.**

Algoritmo 1: `ABRACADABRA`

Entrada: $A[0, 1, \ldots, n-1]$; f (posição final do subvetor de A).
Saída : o produto de todos elementos pertencentes a A.

```
1 se f = 0 então
2 |  return A[0]
3 senão
4 |  return A[f] · ABRACADABRA(A, f − 1)
```

Caso base:

Se n = 1, então f = n-1 = 0, então o vetor tem exatamente um elemento na posição A[0] e ele é o resultado do produto esperado. CQD(Base)

Hipótese indutiva:

Supomos que o problema pode ser resolvido para um vetor de tamanho k.

Passo indutivo:

Queremos provar que o problema pode ser resolvido para um vetor de tamanho k+1.

Seja V um vetor de tamanho k+1. Verifique se a última posição do vetor é igual a posição inicial, se sim, retorne a posição inicial. Senão, o produto dos elementos do vetor V pode ser expresso como o produto entre o elemento da última posição por um subvetor V', construído com os k+1-1 elementos de V e portanto de tamanho igual a k. Pela hipótese de indução, sabemos que é possível resolver o problema para um subvetor do tamanho de V'. Portanto, sempre podemos resolver o mesmo problema para subvetores do vetor V. CQD(Passo)

9. Sabendo que o algoritmo recursivo `AVADAKEDAVRA` tem como resultado o menor inteiro x tal que x pertence a A (fazendo, na primeira chamada, $f = n - 1$ e a igual a qualquer elemento de A), **prove a sua corretude utilizando indução.**

Algoritmo 2: `AVADAKEDAVRA`

Entrada: $A[0, 1, \ldots, n-1]$; f (posição final do subvetor de A);
a (um elemento qualquer que pertence a A).
Saída : a (o menor dos elementos pertencentes a A).

```
1 se f = 0 então
2 |  return min(a, A[0])
3 senão se A[f] < a então
4 |  a ← A[f]
5 return AVADAKEDAVRA(A, f − 1, a)
```

Caso base:

Se n = 1, então f = 0, o vetor A possui apenas um elemento. Portanto a variável a recebe o menor e único valor que está no vetor, A[0]. CQD(Base)

Hipótese indutiva:

Supomos que o problema pode ser resolvido para um vetor de tamanho k.

Passo indutivo:

Queremos provar que o problema pode ser resolvido para um vetor de tamanho k+1.

Seja V um vetor de tamanho k+1. Verificamos se o último elemento do vetor está na primeira posição, se sim, retornamos o valor na primeira posição de V. Senão, verificamos se o elemento na última posição de V é menor que o elemento que está na variável a, se sim, atribuímos o menor valor à a. Em seguida, basta verificar novamente essas condições, para um subvetor V' de tamanho k+1-1 = k. Pela hipótese de indução, o problema pode ser resolvido para vetores de tamanho k. Portanto, podemos encontrar o mínimo, analisando qualquer subvetor do vetor V. CQD(Passo)

10. Assuma que A está ordenado de maneira não-decrescente e que A não contém elementos repetidos. Seja x um elemento de A. O algoritmo recursivo `ALOHOMORA` encontra a posição de x em A (fazendo $i = 0$ e $f = n - 1$ na primeira chamada). **Prove por indução completa** que o algoritmo está correto.

Algoritmo 3: `ALOHOMORA`

Entrada: $A[0, 1, \ldots, n-1]$ (vetor ordenado); i (posição inicial do subvetor de A); f (posição final do subvetor de A); x (elemento pertencente a A).
Saída : m (posição do elemento x em A).

```
1 m ← ⌈(i + f)/2⌉
2 se x = A[m] então
3 |  return m
4 senão se x < A[m] então
5 |  return ALOHOMORA(A, i, m − 1, x)
6 senão se x > A[m] então
7 |  return ALOHOMORA(A, m + 1, f, x)
```

Caso base:

Se o vetor possui apenas um elemento, i = 0 e f = 0, portanto m = 0 é a posição do elemento que procuramos, m é retornado. CQD(Base)

HIpótese indutiva:

Supomos que o problema pode ser resolvido para um vetor de tamanho k, com 1 ≤ k ≤ n.

Passo indutivo:

Seja V um vetor de tamanho k ≥ 2. Primeiro, verificamos se o elemento da posição que procuramos é menor que o elemento da posição central de V. Se sim, resolvemos o mesmo problema para um subvetor V', construído com os elementos da metade esquerda de V. Se não, resolvemos o mesmo problema para um subvetor V'', construído com os elementos da metade direita de V. Pela hipótese de indução, podemos resolver o problema para qualquer vetor de tamanho 1 ≤ k ≤ n. Portanto, é possível dividir sucessivamente o vetor V em partes menores e utilizar o método de divisão e conquista, para se obter a posição do valor procurado. CQD(Passo)