Lista 1 – Processamento da Informação – 2020.1

Universidade Federal do ABC

Aritanan Gruber

aritanan.gruber@ufabc.edu.br

http://professor.ufabc.edu.br/~aritanan.gruber

Entregue os exercícios sinalizados por [$\star$] via Tidia-4 até às 10/03/2018 às 09h00.

1. *Anos bissextos.* No calendário Gregoriano, um ano a é *bissexto* — o mês de Fevereiro tem 29 dias, em vez de 28 — se a é divisível por 4 mas não por 100, ou se a é divisível por 400. Escreva uma função que recebe um inteiro a e devolve verdadeiro se a corresponde a um ano bissexto, ou falso em caso contrário.

2. [$\star$] *Dia da semana.* Escreva uma função que recebe três inteiros d, m, e a, representando dia ($1 \leq d \leq 31$), mês ($1 \leq m \leq 12$) e ano ($a \geq 1$) e devolve o dia da semana em que a data informada ocorreu ou ocorrerá. Para tal, utilize as fórmulas apropriadas para o calendário Gregoriano:
$$\begin{aligned} a_0 &= a - (14 - m)//12 \\ x &= a_0 + a_0//4 - a_0//100 + a_0//400 \\ m_0 &= m + 12 * ((14 - m)//12) - 2 \\ d_0 &= (d + x + (31 * m_0)//12) \ \% \ 7 \end{aligned}$$
Considere que d_0 igual a 0 representa "Domingo", igual a 1, "Segunda", e assim sucessivamente.

3. *Identidade de Bézout.* Escreva uma função que recebe dois inteiros positivos m e n, e devolve três inteiros d, a e b tais que d é o máximo divisor comum a m e n e $am + bn = d$.

4. *Aproximação de* arcsin. Dados $|x| \leq 1$ e $0 < \varepsilon \ll 1$ reais, calcular uma aproximação para $\arcsin x$ (o arco cujo seno é x) através da série infinita de Taylor-Maclaurin ao redor de 0:
$$\arcsin x = x - \frac{x^3}{6} + \frac{3x^5}{40} + \ldots + \frac{(2k)!}{4^k (k!)^2 (2k+1)} x^{2k+1} + \ldots$$
com precisão ε, isto é, até que a diferença absoluta de dois termos consecutivos seja menor que ε:
$$\left| \frac{(2k)!}{4^k (k!)^2 (2k+1)} x^{2k+1} - \frac{(2k-2)!}{4^{k-1} ((k-1)!)^2 (2k-1)} x^{2k-1} \right| < \varepsilon.$$

5. [$\star$] *Binômios de Newton.* Escreva uma função que recebe um inteiro $n \geq 0$ e imprime a expansão do binômio $(x+y)^n$. Por exemplo, para $n = 3$, $(x+y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$ e a saída esperada é: `x**3 + 3*x**2*y + 3*x*y**2 + y**3`. Para tal, utilize a fórmula de expansão do binômio de Newton:
$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i = x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n-1} x y^{n-1} + y^n.$$

6. *Triângulos.* Escreva uma função que recebe três valores a, b e c e determina se eles são lados de um triângulo. Em caso afirmativo, classifique o triângulo que eles formam como: 1 - escaleno, 2 - isósceles, 3 - equilátero. Em caso negativo, devolva 0.

7. [$\star$] *Busca sequencial.* Considere a variante abaixo da busca sequencial, que recebe uma lista $A[0..n-1]$ de inteiros e um inteiro x e promete devolver um índice j tal que $A[j] =$ se $x \in A$, ou $j = n$ em caso contrário. Esta variante faz o prometido? Como? Qual o papel das linhas 2 e 5?

```
  def busca_sequencial (A: [int], x: int) -> int:
    A.append (x)
    j = 0
    while A[j] != x: j += 1
    A.pop()
    return j
```

8. [$\star$] *Embaralhamento?* Considere a seguinte antítese do problema de ordenação: fazer um embaralhamento aleatório dos elementos de uma lista $A[0..n-1]$, ou seja, rearranjar os elementos da lista de modo que todas as permutações sejam igualmente prováveis. Discuta e critique a seguinte solução (cuja variante foi usada na distribuição europeia do Windows 7 da Microsoft).

```
import random

def random_insert (A: [int]):
  for j in range(1, len (A)):
    x = A[j]
    for i in range (j-1,-1,-1):
      if random.random() > 0.5:
        A[i+1] = A[i]
      else:
        break
    A[i+1] = x
```

9. [$\star$] *Busca binária.* Considere o algoritmo abaixo que recebe um inteiro x e uma lista crescente $v[0..n-1]$ e devolve um índice d em $\{0, 1, \ldots n\}$ tal que $v[d-1] < x \leq v[d]$.

```
  def busca_binaria (x: int, v: [int]) -> int:
    e, d = -1, len(v)
    while e < d-1:
      m = (e + d)//2
      if v[m] < x:
        e = m
      else:
        d = m
    return d
```

Responda as seguintes perguntas sobre a função `busca_binaria`. O que acontece se

(a) `while e < d-1:` for trocado por `while e < d:` ou por `while e <= d-1:`?

(b) trocarmos `(e + d)//2` por `(e + d - 1)//2` ou por `(e + d + 1)//2` ou por `(d - e)//2`?

(c) `if (v[m] < x)` for trocado por `if (v[m] <= x)`?

(d) `e = m` for trocado por `e = m+1` ou por `e = m-1`?

(e) `d = m` for trocado por `d = m+1` ou por `d = m-1`?

10. *Inversões.* Seja $A[0..n-1]$ uma lista de $n \geq 0$ números distintos. Se $0 \leq i < j < n$ e $A[i] > A[j]$, então o par (i, j) é chamado de uma *inversão* em A. Por exemplo, se $A = [2, 3, 8, 6, 1]$, os cinco pares $[(0, 4),\ (1, 4),\ (2, 3),\ (2, 4),\ (3, 4)]$ são inversões em A. Escreva uma função que recebe A e devolve uma lista contendo as inversões existentes em A.