

CIn-UFPE

Curso: Ciência da Computação

Disciplina: Paradigmas de Linguagens Computacionais

Docente: Márcio Lopes Cornélio (m l c 2)

Lista de exercícios

Linguagem Lua

1. Uma tabela de tabelas pode ser usada como uma base de dados simples. Por exemplo, para manter informações sobre cientistas famosos, pode-se usar algo assim

```
{ clattes =      {nome = "Cesar",
                  sobrenome = "Lattes",
                  ano_nascimento = 1924,
                  ano_falecimento = 2005,
                  area = "fisica",
                  contribuicao = "meson pi"},
  jpalis =      {nome = "Jacob",
                 sobrenome = "Palis",
                 ano_nascimento = 1940,
                 ano_falecimento = 0000,
                 contribuicao = "sistemas dinamicos"}}
```

Escreva uma função que retorna uma tabela com as chaves usadas em qualquer tabela "interna". No exemplo acima, a função deve retornar {"nome", "sobrenome", "ano_nascimento", "ano_falecimento", "area", "contribuicao"}.

2. Escreva uma outra função, chamada *tabConsistencia*, que recebe uma tabela de tabelas no formato descrito na questão anterior e retorna "verdadeiro" se e somente se todas as tabelas internas possuem as mesmas chaves. Esta função retornaria "falso" para o exemplo da questão anterior, pois a entrada para Jacob Palis não possui a chave "area".

3. Um *vetor esparso* é um vetor que tem quase todas as entradas como zero, como [0,1,0,0,0,0,3,0,0,0]. O armazenamento de todos os zeros desperdiça espaço de memória. Podemos utilizar tabelas para manter apenas as entradas que não são zero. Por exemplo, o vetor anterior pode ser representado pela tabela {[2]=1, [7]=3}.

a. Escreva uma função que toma dois vetores esparsos armazenados como tabelas e retorna uma nova tabela representando a soma deles . (A soma de dois vetores é a soma elemento a elemento, por exemplo, a soma dos vetores [1,2,3] e [4,5,6] é o vetor [5,7,9]).

b. Escreva uma outra função que calcula o produto interno de dois vetores esparsos. (O produto interno de dois vetores é dado pela soma dos produtos do elementos correspondentes dos vetores. Por exemplo, para os vetores [1,2,3] e [4,5,6], temos $4 + 10 + 18 = 32$).

4. Número triangulares (https://en.wikipedia.org/wiki/Triangular_number) constituem uma sequência de inteiros em que o primeiro número é 1. O n -ésimo número triangular é encontrado tomando o número anterior e adicionando n . Escreva uma função (*closure*) para números triangulares. Exemplo

```
for x in num_triangu(6) do
    io.write(x, " ")
end
-- imprime: 1 3 6 10 15 21
```