

## Dünnbesetzte Vektoren und Matrizen als Dictionary speichern

Nehmen Sie an, Sie möchten einen Vektor  $x \in \mathbb{R}^n$  als Dictionary in der Form `vec[i] = value` speichern. Das könnte sinnvoll sein, wenn  $n$  sehr groß ist, aber der Vektor nur wenige Werte ungleich 0 enthält.

1. Schreiben Sie eine Funktion `setValue(vec, i, value)` ohne Rückgabewert, die das Dictionary `vec`, einen Index `i` und einen Wert `value` entgegennimmt und den Vektor `vec` entsprechend verändert.
2. Überlegen Sie sich wie man eine Matrix  $A \in \mathbb{R}^{n \times n}$  als langen Vektor  $x \in \mathbb{R}^{n^2}$  auffassen kann und schreiben Sie eine entsprechende Funktion `setValue(mat, i, j, value)`.
3. Schreiben Sie eine Funktion `value = getValue(mat, i, j)`, die eine dünnbesetzte Matrix `mat` und Indizes `i, j` entgegennimmt und den Matrixeintrag an der Stelle `[i, j]` zurückgibt

*Hinweis:* Ein Schlüssel-Wort Paar sollte die Dimension des Vektors bzw. der Matrix speichern.

**Solution:**

```

1 #!/usr/bin/env python
2 # coding: utf-8
3 # <h1>Table of Contents<span class="tocSkip"></span></h1>
4 # <div class="toc"><ul class="toc-item"><li><span><a href="#Sparse-Vector" data-toc-modified-
   id="Sparse-Vector-1"><span class="toc-item-num">1&nbsp;&nbsp;&nbsp;</span>Sparse Vector</a></span>
   </li></ul></div>
5 # ### Sparse Vector
6 # 1. Sparse Vektor
7 def setValue(vec, i, value):
8     """
9     Sets the value of a vector in a dictionary format.
10    :param vec: dict
11    :param i: int, Index of value
12    :param value: float, Value to be set
13    """
14    if i < vec['dim']:
15        vec[i] = float(value)
16    else:
17        raise ValueError("Index i out of range")
18 vec = {'dim': 92342, 1: 1.43, 14: 0.24, 92341: 3.0}
19 setValue(vec, 19, 1.0)
20 print(vec)
21 setValue(vec, 92342, 1.0)
22 print(vec)
23 # 2. Sparse Matrix
24 def setValue(mat, i, j, value):
25     """
26     Sets the value of a matrix in a dictionary format.
27     :param mat: dict
28     :param i: int, Index of value
29     :param value: float, Value to be set
30     """
31     n, m = mat['dim']
32     if (i < n) and (j < m):
33         mat[n*i + j] = float(value)
34     else:
35         raise ValueError("Index i out of range")
36
37 mat = {'dim': [234, 234], 1: 1.43, 14: 0.24, 2807: 3.0}
38 setValue(mat, 19, 2, 1.0)
39 print(mat)

```

```

40 setValue(mat, 12, 12, 1.0)
41 print(mat)
42 def getValue(mat, i, j):
43     """
44     Sets the value of a matrix in a dictionary format.
45     :param mat: dict
46     :param i: int, Index of value
47     :param value: float, Value to be set
48     """
49     n, m = mat['dim']
50     if (i < n) and (j < m):
51         return mat.get(n*i + j, 0)
52     else:
53         raise ValueError("Index i out of range")
54 getValue(mat, 12, 12)
55 getValue(mat, 12, 13)
56 getValue(mat, 234, 234)

```