

Aufgabe 25:

Sei M eine antisymmetrische $n \times n$ -Matrix, d.h., für alle $i, j \in \{1, \dots, n\}$ gilt: $M_{ij} = -M_{ji}$. Jeder Eintrag sei eine Zahl vom Typ `double`.

- (a) Welche Werte haben die Elemente auf der Hauptdiagonalen, d.h., die Werte M_{ii} für $1 \leq i \leq n$?

Wegen der Antisymmetrie von M gilt: $M_{ij} = -M_{ji}$ für alle $i, j \in \{1, \dots, n\}$.

Für die Matrixelemente der Hauptdiagonalen gilt $i = j$, sodass $\forall i \in \{1, \dots, n\}: M_{ii} = -M_{ii}$, wobei M_{ii} vom Typ `double` ist. Deshalb bleibt nur: $\forall i \in \{1, \dots, n\}: M_{ii} = 0$.

- (b) Wie kann M in einem Array `double[] a` gespeichert werden, sodass die Länge c des Array minimal (speziell kleiner als $n \cdot n$) ist? Wie groß ist c in Abhängigkeit von n ?

Wie in Teil (a) gesehen, sind die Elemente der Hauptdiagonalen einer antisymmetrischen Matrix immer $=0$, sodass die Matrixelemente $M_{ii}, i \in \{1, \dots, n\}$ nicht gespeichert werden müssen. Wegen $M_{ij} = -M_{ji}, \forall i, j \in \{1, \dots, n\}$, reicht es, entweder den Bereich oberhalb oder unterhalb der Hauptdiagonalen zu speichern. Wir entscheiden uns für den unteren Bereich und brauchen nur die Elemente $M_{ij}, \forall i, j \in \{1, \dots, n\}$ mit $i > j$ zu speichern.

Die Länge c des Array `double[] a` beträgt $(n^2 - n)/2$, weil das Array nur die **Hälfte** der $n \cdot n = n^2$ Elementen einer quadratischen Matrix M **ohne** die n Elemente (Nullen) der Hauptdiagonalen enthält.

Die Matrixelemente können in der angegebenen Reihenfolge gespeichert werden:

$$M = \{M_{(2,1)}, M_{(3,1)}, M_{(3,2)}, M_{(4,1)}, M_{(4,2)}, \dots, M_{(n,1)}, M_{(n,2)}, \dots, M_{(n,n-1)}\}.$$

M_{ij} befindet sich somit an Indexposition:

$$p = \sum_{k=1}^{i-2} k + j - 1 = \sum_{k=0}^{i-1} k - i - (i-1) + j - 1 = \frac{i \cdot (i+1)}{2} - 2i + j = \frac{i^2 - 3i}{2} + j$$

Hinweis: $\sum_{k=0}^n k = \frac{n \cdot (n+1)}{2}$ (Arithmetische Reihe)

Die Methode `p(i,j)` bestimmt den absoluten Index p des Elements M_{ij} in dem Array `a[]`:

```
public p(int i,int j) { return (i*i-3*i)/2+j; }
```

Die Methode `mij(p)` bestimmt die Indizes i, j anhand des Index p des Elements M_{ij} in dem Array `a[]`:

```
public int[] mij(int p) {
    int i,j; for (i=2; (i*i-3*i)/2<=p; ++i) ;
    --i; j=p-(i*i-3*i)/2; return new int[]={i,j};
}
```

- (c) Implementieren Sie Ihre Repräsentation durch eine Klasse `MyMatrix` mit einer Methode `public double get(int i,int j)`, die als Rückgabewert M_{ij} liefert. Verwenden Sie Exceptions.

```
public class MyMatrix {
    // private Attribute der Klasse
    private int[] a;
    private int size;

    // Konstruktoren der Klasse
    public MyMatrix() { this.a=new int[]; this.size=0; }
    public MyMatrix(int size) { this.a=new int[size]; this.size=size; }

    // get()-, set()-Methoden für den direkten Zugriff auf das Array.
    public int getA(int index) { return this.a[index]; }
    public void setA(int index,int value) { this.a[index]=value; }

    // get()-, set()-Methoden für den Zugriff auf das Array über Indizes (i,j).
    public int getA(int i,int j) { return this.a[(i*i-3*i)/2+j]; }
    public void setA(int i,int j,int value) { this.a[(i*i-3*i)/2+j]=value; }

    // Methode für das Auslesen eines Elements an der Stelle (i,j) der Matrix M.
    public double M(int i,int j) throws Exception
    {
        if (i>n||i<1) throw new Exception("ungültiger Index i");
        if (j>n||j<1) throw new Exception("ungültiger Index j");
        return (i==j?0 // wenn i=j, dann gebe 0 zurück.
        :(i>j?a[(i*i-3*i)/2+j] // ist i>j?, so gebe a[(i*i-3*i)/2+j],
        :-a[(j*j-3*j)/2+i]); // ansonsten -a[(j*j-3*j)/2+i] zurück.
    }
}
```