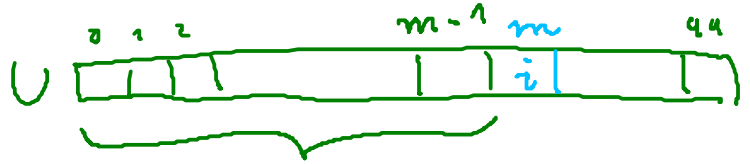


```

class SmallIntSet
{
    static int maxsize=100;
    int m;
    int U[];

    SmallIntSet()
    {
        m = 0;
        U = new int[maxsize];
    }

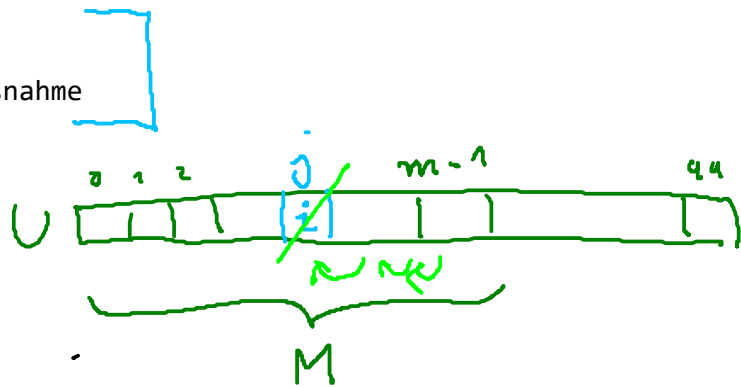
```



```

public boolean add(int i) throws RuntimeException
{
    for (int j=0; j<m; j++)
    {
        if (U[j]==i) return false;
    }
    if (m==maxsize)
        throw new RuntimeException();
    // oder eine spezifischere Ausnahme
    U[m] = i;
    m = m+1;
    return true;
}

```



```

public boolean remove(int i)
{
    for (int j=0; j<m; j++) {
        if (U[j]==i) {
            for (int k=j+1; k<m; k++)
                U[k-1]=U[k];
            m = m-1;
            return true;
        }
    }
    return false;
}

```

Schleifeninvariante

$$\left[ \begin{array}{l}
 U[0], \dots, U[j-1] = U^{alt}[0], \dots, U^{alt}[j-1] \\
 \text{von } i \text{ verschieden} \\
 U^{alt}[j] = i \\
 U[j], \dots, U[k-2] = U^{alt}[j+1], \dots, U^{alt}[k-1] \\
 U[k], \dots, U[m-1] = U^{alt}[k], \dots, U^{alt}[m-1]
 \end{array} \right.$$

```

public boolean contains(int i)
{
    for (int j=0; j<m; j++) {
        if (U[j]==i) return true;
    }
    return false;
}

```

```

}

```