Aufgabe 1

(a) Entscheidungsproblem:

Eingabe: A, x

Ausgabe: *True*, wenn $x \in A$
False, sonst

(b) Berechnungsproblem:

Eingabe: A, x

Ausgabe: Alle i mit $A[i] = x$

Aufgabe 2

0	1	2	3	4	5	6
4	8	9	13	16	20	27

$i = 0,\ j = 6 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 3;\ A[3] = 13 > 9 \Rightarrow j = m - 1 = 2$

$i = 0,\ j = 2 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 1;\ A[1] = 8 < 9 \Rightarrow i = m + 1 = 2$

$i = 2,\ j = 2 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 2;\ A[2] = 9 = 9 \Rightarrow 9 \in A$ (Ende)

Aufgabe 3

0	1	2	3	4	5	6	7
2	7	10	14	19	23	26	31

$i = 0,\ j = 7 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 3;\ A[3] = 14 < 16 \Rightarrow i = m + 1 = 4$

$i = 4,\ j = 7 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 5;\ A[5] = 23 > 16 \Rightarrow j = m - 1 = 4$

$i = 4,\ j = 4 \Rightarrow m = \left\lfloor \frac{i+j}{2} \right\rfloor = 4;\ A[4] = 19 > 16 \Rightarrow j = m - 1 = 3$

die untere Grenze $i = 4$ ist grösser als die obere Grenze $j = 3 \Rightarrow 16 \notin A$

Aufgabe 4

(a) *Best Case:* Das gesuchte Element befindet sich in der Listenmitte und wird sofort gefunden $\Rightarrow O(1)$

(b) *Worst Case:* Das gesuchte Element befindet sich nicht in der Liste oder an der zuletzt getesteten Position $\Rightarrow O(\log n)$

Aufgabe 5

(a) Entscheidungsproblem:

Eingabe: Text t und Muster p

Ausgabe: *True*, wenn p ein Substring (Teilstring) von t ist
False sonst

(b) Berechnungsproblem:

Eingabe: Text t und Muster p

Ausgabe: alle Anfangsposition von p als Substring in t

Aufgabe 6

0	1	2	3	4	5	6	7
T	E	R	R	A	S	S	E

T	E	R	A	S	*
8	8	8	8	8	8
7	8	8	8	8	8
7	6	8	8	8	8
7	6	5	8	8	8
7	6	4	8	8	8
7	6	4	3	8	8
7	6	4	3	2	8
7	6	4	3	1	8

Die unterste Zeile enhält die Verschiebungen zu den Zeichen in der obersten Zeile.

Aufgabe 7

Das Muster steht unmittelbar am Textanfang; also wird das Muster nach m Vergleichen erkannt. Laufzeitkomplexität: $O(m)$

Aufgabe 8

Aufgrund der Voraussetzung kann das Suchmuster bei jedem Vergleich um m Positionen verschoben werden. Daher gilt $O(n/m)$.

Aufgabe 9

Naive Methode:

Y	A	B	B	A	D	A	B	B	A	D	O	O	Vergleiche
A	D	A											1
	A	D	A										2
		A	D	A									1
			A	D	A								1
				A	D	A							3
													8

Boyer-Moore-Horspool:

A	D	*
2	1	3

Y	A	B	B	A	D	A	B	B	A	D	O	O	Vergleiche
A	D	A											1
			A	D	A								1
				A	D	A							3
													5

Aufgabe 10

```
def linearsearch(L, item):
    for x in L:
        if x == item:
            return True
    return False
```