

1.

Algorithmus

Input ein Queue q der Länge n aus Buchstaben

Output boolean

```

 $\mathcal{O}(1)$  1: if  $n \% 2 == 0$ 
 $\mathcal{O}(1)$  2:     is_even = True
 $\mathcal{O}(1)$  3:      $m = n / 2$ 
/      4: else
 $\mathcal{O}(1)$  5:     is_even = False
 $\mathcal{O}(1)$  6:      $m = (n - 1) / 2$ 
/      7: end if
 $\mathcal{O}(1)$  8: Stack  $s[m]$  # initialisiere ein Stack der Länge  $m$ 
 $\mathcal{O}(n)$  9: for  $i=1$  to  $m$  do
 $\mathcal{O}(1)$  10:     $x = q.head()$ 
 $\mathcal{O}(1)$  11:     $s.push(x)$ 
 $\mathcal{O}(1)$  12:     $q.dequeue()$ 
/      13: end for
 $\mathcal{O}(1)$  14: if not is_even
 $\mathcal{O}(1)$  15:     $q.dequeue()$ 
/      16: end if
 $\mathcal{O}(n)$  17: for  $i=1$  to  $m$  do
 $\mathcal{O}(1)$  18:    if  $q.head() != s.top()$ 
 $\mathcal{O}(1)$  19:        return False
/      20:    end if
 $\mathcal{O}(1)$  21:     $q.dequeue()$ 
 $\mathcal{O}(1)$  22:     $s.pop()$ 
/      23: end for
 $\mathcal{O}(1)$  24: return True

```

Begründung: Vergleiche oben für die Laufzeitanalyse zeilenweise. Aber auf jeden Fall lesen wir die Eingabe nur einmal ab (die erste Hälfte ab Zeile 9, die zweite Hälfte ab Zeile 17), daher ist die Laufzeit $\mathcal{O}(n)$.

2.

Algorithmus Dat Kann Doch Net Wahr Sein

Input ein Queue q der Länge n aus Buchstaben

Output boolean

```

1: a = 0
2: b = 0
3: c = 0
4: d = 0
5: e = 0
6: f = 0
7: g = 0
8: h = 0
9: i = 0
10: j = 0
11: k = 0
12: l = 0
13: m = 0
14: n0 = 0
15: o = 0

```

```
16: p = 0
17: q0 = 0
18: r = 0
19: s = 0
20: t = 0
21: u = 0
22: v = 0
23: w = 0
24: x = 0
25: y = 0
26: z = 0
27: for indx = 1 to n do
28:   if q.head() == "a"
29:     a += 1
30:   end if
31:   if q.head() == "b"
32:     b += 1
33:   end if
34:   if q.head() == "c"
35:     c += 1
36:   end if
37:   if q.head() == "d"
38:     d += 1
39:   end if
40:   if q.head() == "e"
41:     e += 1
42:   end if
43:   if q.head() == "f"
44:     f += 1
45:   end if
46:   if q.head() == "g"
47:     g += 1
48:   end if
49:   if q.head() == "h"
50:     h += 1
51:   end if
52:   if q.head() == "i"
53:     i += 1
54:   end if
55:   if q.head() == "j"
56:     j += 1
57:   end if
58:   if q.head() == "k"
59:     k += 1
60:   end if
61:   if q.head() == "l"
62:     l += 1
63:   end if
64:   if q.head() == "m"
65:     m += 1
66:   end if
67:   if q.head() == "n"
68:     n0 += 1
69:   end if
70:   if q.head() == "o"
71:     o += 1
```

```

72:     end if
73:     if q.head() == "p"
74:         p += 1
75:     end if
76:     if q.head() == "q"
77:         q0 += 1
78:     end if
79:     if q.head() == "r"
80:         r += 1
81:     end if
82:     if q.head() == "s"
83:         s += 1
84:     end if
85:     if q.head() == "t"
86:         t += 1
87:     end if
88:     if q.head() == "u"
89:         u += 1
90:     end if
91:     if q.head() == "v"
92:         v += 1
93:     end if
94:     if q.head() == "w"
95:         w += 1
96:     end if
97:     if q.head() == "x"
98:         x += 1
99:     end if
100:    if q.head() == "y"
101:        y += 1
102:    end if
103:    if q.head() == "z"
104:        z += 1
105:    end if
106:    q.dequeue()
107: end for
108: count = 0
109: if a % 2 != 0
110:     count += 1
111: end if
112: if b % 2 != 0
113:     count += 1
114: end if
115: if c % 2 != 0
116:     count += 1
117: end if
118: if d % 2 != 0
119:     count += 1
120: end if
121: if e % 2 != 0
122:     count += 1
123: end if
124: if f % 2 != 0
125:     count += 1
126: end if
127: if g % 2 != 0

```

```
128:     count += 1
129: end if
130: if h % 2 != 0
131:     count += 1
132: end if
133: if i % 2 != 0
134:     count += 1
135: end if
136: if j % 2 != 0
137:     count += 1
138: end if
139: if k % 2 != 0
140:     count += 1
141: end if
142: if l % 2 != 0
143:     count += 1
144: end if
145: if m % 2 != 0
146:     count += 1
147: end if
148: if n0 % 2 != 0
149:     count += 1
150: end if
151: if o % 2 != 0
152:     count += 1
153: end if
154: if p % 2 != 0
155:     count += 1
156: end if
157: if q0 % 2 != 0
158:     count += 1
159: end if
160: if r % 2 != 0
161:     count += 1
162: end if
163: if s % 2 != 0
164:     count += 1
165: end if
166: if t % 2 != 0
167:     count += 1
168: end if
169: if u % 2 != 0
170:     count += 1
171: end if
172: if v % 2 != 0
173:     count += 1
174: end if
175: if w % 2 != 0
176:     count += 1
177: end if
178: if x % 2 != 0
179:     count += 1
180: end if
181: if y % 2 != 0
182:     count += 1
183: end if
```

```
184: if z % 2 != 0
185:     count += 1
186: end if
187: if count > 1
188:     return false # with special thanks to my girlfriend
189: else
190:     return true # or the Python script I wrote
191: end if
```

Begründung: Jede Zeile im obigen Code wird in $\mathcal{O}(1)$ ausgeführt bis auf die eine for-Schleife, die einmal die Eingabe einliest, was $\mathcal{O}(n)$ ist. Daher ist die gesamte Laufzeit auch $\mathcal{O}(n)$.