

23. Bestimmen der Mehrheit, 10 Punkte

a) nix, weil 0 Punkte. :)

b) Idee:

Liste durchlaufen, Histogramm(Wert, Anzahl) erstellen, Histogramm durchlaufen, wenn Wert Bedingung B erfüllt ausgeben

```
1 in: e[0,...,i]           ; Werte
2 var: i = 0               ; Laufvariable
3 array: hist = []         ; Leeres Array, mit 0 initialisiert
4 var: n = len(a)          ; laenge von Eingabe
5 var: B = (n+1)/2 + 1     ; Abschlussbedingung
6 ; Histogramm befüllen
7 for i=0, i<n, i++:
8     ;eingelesener Wert entspricht der Position im Histogramm
9     hist[e[i]]++
10 ; Histogramm durchlaufen
11 for val in hist:
12     ; Wert erfüllt Bedingung
13     if val >= B:
14         ; Wert zur Ausgabe hinzufügen
15         out.append(val)
16 return out
```

Laufzeit

Argumentativ: $\mathcal{O}(c * n)$, mit n = Länge der Eingabe.

Zeilen 1-5: konstante Laufzeit

Zeile 7-9: $\mathcal{O}(n)$

Zeile 11-15: $\mathcal{O}(l(hist))$

Obere und untere Schranke unterscheiden sich nicht wesentlich und sind abhängig in der Größe vom Histogramm. Allgemein reicht es allerdings aus sich die Eingabe nur einmal $\Theta(n)$ anzusehen. Im schlimmsten Fall (alles verschiedene Werte) beträgt die Laufzeit $\mathcal{O}(2n)$.

Im schlimmsten Fall könnte man die Laufzeit optimieren in dem die Größe (Anzahl der Einträge) des Histogramms im Bezug zur Bedingung betrachtet.

c) Nehme Algorithmus aus b) und wähle $B = n * 0.33$.

25. Lokales Maximum in Bäumen, 10 Punkte