

Übungen zur Algorithmischen Bioinformatik I

Blatt 2

Xiheng He

April 2021

1. Aufgabe (10 Punkte):

```
FIND(int[] a, int x, int start, int end)
```

```
begin
    int m := ⌊ $\frac{start+end}{2}$ ⌋;
    if (a[m] > x) then
        return FIND(a, x, start, m);           /* If a[m] is greater than x then return the left half */
    if (a[m] < x) then
        return FIND(a, x, m + 1, end);         /* If a[m] is smaller than x then return the right half */
    else
        return m;                               /* a[m] == x and m is the position */
end
```

Analyse:

Eingabegröße: $|A| = n$.

Charakteristischen Operationen: $>$ und $<$.

Laufzeit: Vergleich braucht einmal Pro Rekursion. Somit ist die Laufzeit:

$$A_{DC}(n) = \begin{cases} 1 & \text{falls } n = 1 \\ A_{DC}(\lfloor n/2 \rfloor) + 1 & \text{falls } n > 1 \end{cases}$$

Angenommen, dass $n = 2^k$:

$$A_{DC}(n) = A(n/2) + 1 = A(n/4) + 2 = A(n/8) + 3 = A(2^k/2^k) + k = A(1) + \log(n) = 1 + \log(n)$$

Deshalb: $A_{DC}(n) \in \Theta(\log(n))$.

Korrektheit: Sei A sortiert, es gibt immer nur drei Fälle in jeder rekursion: $x = a[m]$, $x < a[m]$ und $x > a[m]$. Wenn x nicht $a[m]$ gleich ist, wird x in nächster rekursion weiter gesucht. Wenn x kleiner als $a[m]$ ist und nicht in der linken Hälfte gefunden werden kann, ist die linke Hälfte offensichtlich nicht sortiert. Daher widerspricht es und damit ist der Algorithmus korrekt. Analog kann man die rechte Hälfte auch beweisen wenn x größer als $a[m]$.