

1. Feinanalyse (6 Punkte)

Zur Berechnung der n -ten Potenz einer Zahl x (also $p = x^n$) schlägt jemand den folgenden einfachen Algorithmus vor:

```
simplePower (↓x ↓n ↑p ) {  
    p = 1;  
    for(i = 1; i <= n; i++) {  
        p = p * x  
    }  
}
```

Um die Berechnung schneller zu machen, schlägt ein anderer den folgenden komplizierteren Algorithmus vor:

```
quickPower (↓x ↓n ↑p ) {  
    p = 1  
    while(n > 0) {  
        if (n mod 2 != 0) {  
            p = p * x  
        }  
        n = n / 2  
        x = x * x  
    }  
}
```

a) Beschreiben Sie die beiden Verfahren zum Potenzieren. Warum “verdient” der zweite Algorithmus seinen Namen?

b) Führen Sie für beide Algorithmen eine Feinanalyse durch

c) Welcher Algorithmus ist unter welchen Bedingungen schneller

Hinweise: Verwenden Sie für die Berechnung die in der Vorlesung vorgestellten Ausführungszeiten der einzelnen Anweisungen. Die *modulo* Operation benötigt 5 Zeiteinheiten, *++* 1.8 Zeiteinheiten, *Multiplikation* 3 Zeiteinheiten

2. Grobanalyse (4 + 3 + 3 Punkte)

Entwickeln Sie die folgenden Funktionen und analysieren Sie diese infolge.

Bestimmen Sie den Best, Worst und Average-Case (sofern möglich)

`int findCharLeft(const char* str, char ch)` bestimmt die Position des ersten Zeichens von links mit dem Wert `ch` in der Zeichenkette `str`.

`int findCharRight(const char* str, char ch)` bestimmt die Position des ersten Zeichens von rechts mit dem Wert `ch` in der Zeichenkette `str`.

`int findCharRandom(const char* str, char ch)` bestimmt die Position des ersten Zeichens mit dem Wert `ch` in der Zeichenkette `str` anhand zufällig ausgewählter Positionen.

Implementieren Sie die Funktionen in C. Achten Sie auf eine möglichst effiziente Implementierung.

3. O-Notation (4 Punkte)

In der Vorlesung wurden die unterschiedlichen Komplexitätsklassen besprochen. Suchen Sie in „echten“ Quellen für die Komplexitätsklassen $O(1)$, $O(n)$, $O(\log(n))$ und $O(n^2)$ jeweils zwei Algorithmen. Überlegen Sie sich auch eine Begründung für die Einordnung in die jeweilige Klasse.