Prof. Dr. Alfred Schreiber – Institut für Mathematik und ihre Didaktik

"Algorithmen"
Lösung zu Aufgabe 24

Aufgabe 24

Vergleichen Sie die in der Bibliothek **`funclib.js`** hinzugefügte Variante **`PrimeSieve2(n)`** mit der ursprünglichen Fassung. Beachten Sie: Die Hilfsfunktion **`Sieve( p, tab )`** wurde ebenfalls durch eine angepasste Variante **`Sieve2( p, tab )`** ersetzt.

Begründen Sie, weshalb die neue Variante dasselbe Ergebnis auf etwas verbesserte Weise berechnet.

Lösung (siehe dazu die Funktionsbibliothek funclib.js)

Die Variante des Primzahl-Siebs nach Eratosthenes beruht auf der Idee, als Ausgangspunkt nicht die Liste $\{1, 2, \ldots, n\}$ zu nehmen, sondern von vornherein alle geraden Zahlen zu streichen, d.h. die Liste $\{1, 3, 5, \ldots, 2m - 1\}$ zu betrachten, wobei m maximal mit $2m - 1 \leq n$ zu wählen ist. Im Initalisierungteil von **`PrimeSieve2`** geschieht genau dies.

Da 1 zu streichen ist, andererseits 2 (als einzige gerade Zahl) aufzunehmen ist, wird die Liste modifiziert zu $\{2, 3, 5, \ldots, 2m - 1\}$ und als Startzahl $p = 3$ (statt wie früher $p = 2$) gewählt, weil ja mit 2 die Liste nicht gesiebt werden muss.

Die Hauptfunktion **`PrimeSieve2`** unterscheidet sich im Übrigen von **`PrimeSieve`** lediglich darin, dass anstelle von **`Sieve`** die angepasste Funktion **`Sieve2`** in der while-Schleife zum Einsatz kommt. Sie leistet dasselbe wie ihre Vorgängerin (nämlich das Sieben der Liste mittels p).

In der while-Schleife der Funktion **`Sieve2`** muss man nun nicht mehr abfragen, ob p gleich 2 ist. Es ist lediglich zu beachten, dass die Platznummer von p^2 in der aktuellen Liste nicht mehr p^2 ist, sondern $\frac{p^2+1}{2}$ (unmittelbar zu sehen!).

Das Siebverfahren verläuft damit unverändert, aber leicht verbessert dadurch, dass immerhin *ein* Siebschritt (und das damit verbundene Traversieren der kompletten Liste) eingespart wird. Die Größenordnung der Laufzeit wird allerdings hierdurch nicht herabgesetzt.