

Lösungen

Der hier vorgestellte Algorithmus orientiert sich am Algorithmus von Jens Stoye und Dann Gusfield aus dem Paper 'Simple and flexible detection of contiguous repeats using a Suffix Tree', Theoretical Computer Science 270 (2002) 843â€“856

Predefinitions:

Sei $T(S)$ der SuffixTree von String S .

Sei v ein interner Knoten (kein Blatt oder root) von $T(S)$.

Dann ist $LL(v)$ die Liste aller Blatt-Label, welche vom Knoten v aus erreicht werden können. Und $D(v)$ die Länge des Labels (der Kantenbeschriftung $L(v)$) von der Wurzel zu v .

Weiterhin sind alle Knoten unmarkiert.

Der Algorithmus arbeitet wie folgt:

```
FOR v IN T(S) DO
  IF v IS unmarked THEN
    create_LL(v);
    FOR EACH i IN LL(v) DO
      IF i+D(v) IS IN LL(v) THEN
        IF S[i] != S[i+2D(v)] THEN
          REPORT L(v); /*so called branching repeat*/
          WHILE (S[i-1] == S[i+2D(v)+1]) DO
            REPORT S[i-1]; /*non-branching repeat*/
          DONE
        END IF
      END IF
    DONE
    mark(v);
  ELSE
    next(v);
  END IF
DONE
```