

Bereits eingegeben seien: Die Zahl n und n Wahrscheinlichkeiten p_1, p_2, \dots, p_n . Der m -te Wert gibt die Wahrscheinlichkeit für das m -te Element a_m in der *geordneten* Folge $a_1 a_2 a_3 \dots a_n$.
Ausgabe ist der Wert $S[1,n]$. Aus den $R[i,j]$ kann man den optimalen Suchbaum rekonstruieren (Wie? Selbst programmieren!).

```

var i, j, k, m, n, diff: natural; min: real; P: array [1..n] of real;
      S,  $\Gamma$ : array [1..n+1] of array [1..n] of real;
      R: array [1..n] of array [1..n] of natural;
begin
  for i:=1 to n do
    S[i+1,i] := 0.0; S[i,i] := P[i];  $\Gamma$ [i,i] := P[i]; R[i,i]:=i od;
  for diff:=1 to n-1 do
    for i:=1 to n-diff do
      j := i+diff;  $\Gamma$ [i,j] :=  $\Gamma$ [i,j-1] + P[j]; min := S[i+1,j]; m := i;
      for k:=i+1 to j do
        if S[i,k-1] + S[k+1,j] < min then
          min := S[i,k-1] + S[k+1,j]; m := k fi od;
      S[i,j] := min +  $\Gamma$ [i,j]; R[i,j] := m;
    od od
  end.

```

Optimaler Suchbaum in Zeit $\Theta(n^3)$