Lista 6 Zadanie 3, AiSD

9.05.2020

Rozważmy następujący problem. Dla zadanego ciągu wejściowego złożonego z n liczb całkowitych $a[1,\ldots,n]$ należy znaleźć podciąg spójny $a[j,\ldots,j+k]$, którego suma wyrazów jest największa. Poniższy algorytm 1 stanowi rozwiązanie naszego problemu, zwracając na wyjściu szukany podciąg. W konstruowanym algorytmie zakładamy, ze wejście nie może być tablicą długości 0. Przyjmujemy także, że zwrócony ciąg nie może być ciągiem pustym, tzn. musi mieć długość co najmniej 1.

Algorytm 1 Wyznaczanie podciągu spójnego o największej sumie

Input: $a[1,\ldots,n]$ – ciąg wejściowy

Output: $a[j,\ldots,j+k]$ – podciąg spójny ciągu a o największej sumie

1: $c = s = a[1]$ /* *s - maksymalna suma zwróconego podciągu* */
2: $cb = ce = 1$
3: $sb = 1$ /* *sb - początkowy indeks podciągu o maksymalnej sumie* */
4: $se = 1$ /* *se - końcowy indeks podciągu o maksymalnej sumie* */
5: **for** $i = 2$ **to** n **do**
6:     **if** $c > 0$ **then**
7:         $c = c + a[i]$
8:         $ce = i$
9:     **else**
10:         $c = a[i]$
11:         $cb = ce = i$
12:     **if** $c > s$ **then**
13:         $s = c$
14:         $se = ce$
15:         $sb = cb$
16: **return** $a[sb,\ldots,se]$

Złozoność: $T(n) = O(n)$, ponieważ każdy element tablicy wejściowej przetwarzany jest raz, a w trakcie każdej iteracji pętli **for** wykonywana jest stała ilość operacji o koszcie $O(1)$.

Dowód: Oznaczmy przez c_i wartość zmiennej c po wykonaniu pętli **for** (linie 5–15 w algorytmie 1) dla ustalonej wartości $i \in \{2,\ldots,n\}$ i niech $c_1 = a[1]$ oznacza wartość c w momencie pierwszego wejścia do pętli **for**. Udowodnimy, że niezmiennikiem pętli jest $c_i = max(S_{1,i}, S_{2,i}, \ldots, S_{i,i})$ (największa tymczasowa suma dla każdej iteracji pętli), gdzie $S_{k,l} = a[k] + \ldots + a[l]$ dla danego i. Zauważmy, że $c_1 = a[1] = S_{1,1}$. Ustalmy więc $i \in \{2,\ldots,n\}$ i zobaczmy, jak modyfikowana jest wartość zmiennej c w trakcie iteracji pętli **for** dla wartości iteratora i.

Jeśli dotychczasowa wartość $c_{i-1} = c > 0$, to wykonujemy podstawienie $c = c + a[i]$ (linia 7), wobec czego

$$\begin{aligned} c_i = c_{i-1} + a[i] &= \max(S_{1,i-1}, S_{2,i-1}, \ldots, S_{i-1,i-1}) + a[i] \\ &= \max(S_{1,i-1} + a[i], S_{2,i-1} + a[i], \ldots, S_{i-1,i-1} + a[i]) \\ &= \max(S_{1,i}, S_{2,i}, \ldots, S_{i-1,i}) \,. \end{aligned} \tag{1}$$

Zauważmy także, że warunek $c_{i-1} > 0$ implikuje $c_i = c_{i-1} + a[i] > a[i]$. Jeśli natomiast $c_{i-1} \leqslant 0$, wówczas wykonamy podstawienie $c = a[i]$, a zatem $c_i = a[i] \geqslant c_{i-1} + a[i]$. Wobec powyższego wnioskujemy, że wartość zmiennej c_i na końcu iteracji pętli **for** wynosi

$$c_i = \max(c_{i-1} + a[i], a[i]) \,.$$

Wobec faktu, iż $a[i] = S_{i,i}$, ze wzoru (1) otrzymujemy

$$\begin{aligned} c_i &= \max(\max(S_{1,i}, S_{2,i}, \ldots, S_{i-1,i}), S_{i,i}) \\ &= \max(S_{1,i}, S_{2,i}, \ldots, S_{i-1,i}, S_{i,i}) \,, \end{aligned}$$

czego należało dowieść. c_i przechowuje zatem wartość maksymalnej sumy podciągów kończących się na elemencie $a[i]$. Oznaczając przez s_i wartość zmiennej s wyliczanej w liniach 12–13 pętli **for** dla wartości iteratora równej i (dla $i = 1$ mamy $s_1 = a[1]$), nietrudno zauważyć, że

$$s_i = \max(c_1, \ldots, c_i) \,.$$

Tak więc s_n jest wartością sumy podciągu o maksymalnej sumie, a dzięki aktualizowaniu w pętli indeksów se i sb jesteśmy w stanie odtworzyć na koniec szukany podciąg z tablicy a.

Przedstawione rozwiązanie wykorzystuje technikę programowania dynamicznego – w i-tym kroku wyznaczana jest wartość maksymalnej sumy podciągu spójnego kończącego się na elemencie $a[i]$ na podstawie wcześniej obliczonych rozwiązań dla podproblemów mniejszego rozmiaru.