

Najdłuższy wspólny podciąg

Konrad Łakomy

26 05 2014

Opis algorytmu NWP

Najdłuższy wspólny podciąg (NWP, ang. longest common subsequence) jest to najdłuższy podciąg znaków, które występują w tej samej kolejności w dwóch porównywanych łańcuchach. Elementy podciągów nie muszą przy tym leżeć obok siebie (tym różni się ten problem od problemu najdłuższego wspólnego podłańcucha, ang. longest common substring). Rozwiązanie tego problemu jest bardzo przydatne przy pisaniu programów mających na celu wykrycie zmian w dokumentach lub plikach, lub przy pisaniu programów służących do identyfikacji plagiatów.

Algorytm jest przykładem potęgi programowania dynamicznego. Rozwiązanie konstruowane jest stopniowo - zaczynamy od mniejszych (zatem łatwiejszych) podproblemów. Wyniki dla nich pozwolą nam na utworzenie odpowiedzi dla pierwotnego zadania.

Algorytm ten ma złożoność obliczeniową rzędu $O(n * m)$, gdzie n oraz m to długości ciągów A i B . Algorytm ten tworzy tablicę dwuwymiarową C (n na m) taką, że wartość $C[i][j]$ jest długością NWP podciągów $A[1..i]$ i $B[1..j]$. A więc po zakończeniu wypełniania tablicy C komórka $C[n][m]$ będzie zawierała wartość będącą długością NWP oryginalnych ciągów wejściowych A i B .

Przykłady działania algorytmu - Przykład 1

Ciąg pierwszy: ALIBABA

Ciąg drugi: KALIMALBA

Rezultat: ALIABA

Długość: 6

Tablica:

	0	K	A	L	I	M	A	L	B	A
0	0	0	0	0	0	0	0	0	0	0
A	0	0	1	1	1	1	1	1	1	1
L	0	0	1	2	2	2	2	2	2	2
I	0	0	1	2	3	3	3	3	3	3
B	0	0	1	2	3	3	3	3	4	4
A	0	0	1	2	3	3	4	4	4	5
B	0	0	1	2	3	3	4	4	5	5
A	0	0	1	2	3	3	4	4	5	6

Przykłady działania algorytmu - Przykład 2

Ciąg pierwszy: LABRADOR

Ciąg drugi: RABARBAR

Rezultat: ABRAR

Długość: 5

Tablica:

	0	R	A	B	A	R	B	A	R
0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
A	0	0	1	1	1	1	1	1	1
B	0	0	1	2	2	2	2	2	2
R	0	1	1	2	2	3	3	3	3
A	0	1	2	2	3	3	3	4	4
D	0	1	2	2	3	3	3	4	4
O	0	1	2	2	3	3	3	4	4
R	0	1	2	2	3	4	4	4	5

Wnioski

Na podstawie powyższych przykładów możemy stwierdzić iż algorytm NWP działa poprawnie. Jest on przykładem programowania dynamicznego w którym stosuje się technike dzielenia problemu głównego na podproblemy.

Taki sposób realizacji algorytmu możemy wykorzystać w sytuacji gdy podproblemy na które został podzielony problem główny nie są nie zależne, ponieważ w różnych podproblemach wykonywane są wiele razy te same obliczenia które możemy wykorzystać.

Wyniki obliczeń zapamiętywane są w tablicy pomocniczej, która jest wykorzystywana w kolejnych krokach algorytmu, co eliminuje potrzebe wielokrotnego wykonywania tych samych obliczeń. Prowadzi to do zmniejszenia złożoności obliczeniowej.

Programowanie dynamiczne polega więc na wykonania obliczeń każdego podproblemu tylko raz i zapamiętaniu jego wyniku w tabeli. W każdym kolejnym kroku można z tej tabeli korzystać.