Największy wspólny podciąg - Laboratorium nr 10 z PAMSI

Justyna Klijewska

28 05 2014

1 WSTĘP

Celem ćwiczenia było zaimplementowanie algorytmu Ńajwiększy wspólny podciąg"(taki został przeze mnie wylosowany).

2 TEORIA

Algorytm, który wyznacza najdłuższy wspólny podciąg (z ang. LCS - Longest Common Subsequence) dwóch wyrazów, jest przykładem potęgi programowania dynamicznego. Elementy podciągów nie muszą przy tym leżeć obok siebie (tym różni się ten problem od problemu najdłuższego wspólnego podłańcucha, ang. longest common substring).

3 IMPLEMENTACJA

Początkowo pobierane są dane o ilości znaków w naszych podciągach (te z kolei są wczytywane z pliku). Wyświetlana jest ilość wierszy i kolumn tablicy (o jeden większych niż odpowiednie długości wyrazów), która posłuży do rozwiązania algorytmu. Następnie tworzona jest tablica:
- pierwszy rząd i kolumna wypełnione są zerami
- jeżeli A[i]=B[j] to musimy dodać jeden do (i-1, j-1)
- jesli jednak nie są równe to w (i,j) wpisujemy większą z liczb znajdujących się w komórkach (i-1,j) i (i,j-1).
Gdzie A, B -ciągi i, j- lokalizacja z tablicy
Kiedy już to mamy wystarczy wyświetlić nadłuższy wspólny podciąg.

4 PRZYKŁADY

Przykład 1
1 ciąg: abaabbaaa Długość:9
2 ciąg: babab Długość:5

Tablica:

		a	b	a	a	b	b	a	a	a
	0	0	0	0	0	0	0	0	0	0
b	0	0	1	1	1	1	1	1	1	1
a	0	1	1	2	2	2	2	2	2	2
b	0	1	2	2	2	3	3	3	3	3
a	0	1	2	3	3	3	3	4	4	4
b	0	1	2	3	3	4	4	4	4	4

Wynik: abab Długość:4

Przykład 2
1 ciąg: justyna Długość:7
2 ciąg: justa Długość:5

Tablica:

		j	u	s	t	y	n	a
	0	0	0	0	0	0	0	0
j	0	1	1	1	1	1	1	1
u	0	1	2	2	2	2	2	2
s	0	1	2	3	3	3	3	3
t	0	1	2	3	4	4	4	4
a	0	1	2	3	4	4	4	5

Wynik: justa Długość:5

Przykład 3
1 ciąg: co66533 Długość:7
2 ciąg: cos13 Długość:5

Tablica:

		c	o	6	6	5	3	3
	0	0	0	0	0	0	0	0
c	0	1	1	1	1	1	1	1
o	0	1	2	2	2	2	2	2
s	0	1	2	2	2	2	2	2
1	0	1	2	2	2	2	2	2
3	0	1	2	2	2	2	3	3

Wynik: co3 Długość:3

5 WNIOSKI

- Dzięki zastosowaniu dynamicznego programowania możemy zyskać większą efektywność - przez dzielenie problemu na mniejsze problemy.
- W odróżnieniu od techniki dziel i zwyciężaj (którą poznaliśmy przy sortowaniu) w programowaniu dynamicznym problemy nie są rozłączne.
- Złożoność obliczeniowa algorytmu to O(n*m), gdzie n i m to długości ciągów.
- Istnieje wiele sposób "udoskonalenia"tej złożoności obliczeniowej. Przy tak małych ciągach nie będzie miało to większej różnicy, jednak przy większych warto pomyśleć o optymalizacji. Może nią być np.: nie przechowywanie w pamięci niepotrzebnych elementów tablicy (od 0 do i-2). Wtedy nasza złożoność wyniesie O(2*m).
- 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.