

Algorytmy i struktury danych

Wykład 1

wprowadzenie, struktury sterujące, projektowanie algorytmów

prof. dr hab. inż. Andrzej Obuchowicz

Instytut Sterowania i Systemów Informatycznych
Uniwersytet Zielonogórski
a.obuchowicz@issi.uz.zgora.pl
p. 424 A2

22 września 2021

Spis treści

- 1 Wprowadzenie
 - Warunki zaliczenia
 - Zakres tematyczny
 - Plan wykładu
 - Literatura
- 2 Algorytm
 - Problem algorytmiczny
 - Algorytm i jego własności
- 3 Struktury sterujące
 - Schematy blokowe struktur sterujących
 - Przykład - sortowanie bąbelkowe
- 4 Projektownie algorytmów
 - Metoda „dziel i zwyciężaj”
 - Programowanie dynamiczne
 - Metoda zachłanna
- 5 Już za tydzień na wykładzie

Warunki zaliczenia

Wykład: kończy się egzaminem pisemnym obejmującym zagadnienia omawiane na wykładzie. Egzamin odbywa się podczas sesji zimowej.

Zakres tematyczny – Syllabus

- (I) Algorytm i jego własności: pojęcie problemu algorytmicznego i algorytmu, własności algorytmów; struktury sterujące i schematy blokowe. Techniki programowania: rekurencja i derekursywacja, programowanie typu „dziel i rządź”, algorytmy zachłanne, programowanie dynamiczne.
- (II) Struktury danych: pojęcie struktury danych, zbiory dynamiczne, zbiory liniowo uporządkowane, słownik; kolejki i stosy; listy jedno- i dwukierunkowe, listy cykliczne, drzewa binarne, drzewa o dowolnej liczbie potomków; kolejki priorytetowe.

Zakres tematyczny

- (III) Słowniki: drzewa przeszukiwań binarnych BST i AVL, drzewa czerwono-czarne; tablice haszujące, funkcje haszujące, techniki zapobiegania konfliktom; B-drzewa.
- (IV) Zbiory i grafy: zbiory, grafy, reprezentacje grafów, przeszukiwanie wszerz i w głąb, algorytmy teoriografowe i sieciowe.
- (V) Analiza wybranych problemów algorytmicznych: przeszukiwanie liniowe i binarne, wybór k-tego elementu; wewnętrzne i zewnętrzne sortowanie danych; wyszukiwanie wzorca w tekście; algorytmy geometryczne; zagadnienie stronicowania; układy arytmetyczne; podstawowe techniki kompresji i kodowania danych.

Plan wykładu – spotkania tydzień po tygodniu

- (1) wprowadzenie, schematy blokowe, metody budowy algorytmów
- (2) sortowanie wewnętrzne
- (3) sortowanie zewnętrzne
- (4) techniki wyszukiwania wzorca
- (5) elementarne struktury danych
- (6) drzewa przeszukiwań binarnych, AVL
- (7) drzewa czerwono-czarne, struktury samoorganizujące się
- (8) B-drzewa, drzewa wyszukiwań pozycyjnych

Plan wykładu – spotkania tydzień po tygodniu

- (9) haszowanie, kolejki priorytetowe
- (10) algorytmy teorio-grafowe
- (11) algorytmy teorio-liczbowe
- (12) algorytmy geometryczne
- (13) układy arytmetyczne; podstawowe techniki kompresji i kodowania danych
- (14) powtórka do egzaminu
- (15) egzamin - termin 0

Literatura podstawowa

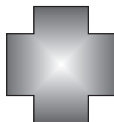
- ❶ Cormen T. H., Leiserson C. E., Rivest R. L.: *Wprowadzenie do algorytmów*, WNT, Warszawa, 1997
- ❷ Kotowski P.: *Algorytmy + struktury danych = abstrakcyjne typy danych*, Wyd. BTC, Warszawa, 2006
- ❸ Wróblewski P.: *Algorytmy, struktury danych i języki programowania*, Helion, Gliwice, 1997
- ❹ Aho A. V., Hopcroft J. E., Ullman J.D.: *Algorytmy i struktury danych*, Helion, Gliwice, 2003

Literatura uzupełniająca

- ① Adamski T., Ogrodzki J.: *Algorytmy komputerowe i struktury danych*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2005
- ② Banachowski L., Diks K., Rytter W.: *Algorytmy i struktury danych*, WNT, Warszawa, 1996
- ③ Harris S., Ross J.: *Od podstaw algorytmu*, Helion, Gliwice, 2006
- ④ Neapolitan R., Naimipour K.: *Podstawy algorytmów z przykładami w C++*, Helion, Gliwice, 2004
- ⑤ Stephens R.: *Algorytmy i struktury danych stosowane w Delphi 3, 4 i 5 z przykładami w Delhi*, Helion, Gliwice, 2000
- ⑥ Wirth N.: *Algorytmy + struktury danych = programy*, WNT, Warszawa, 2002

Sformułowanie problemu

Opis
wszystkich
poprawnych
danych
wejściowych



Opis
oczekiwanych
wyników jako
funkcji danych
wejściowych

Przykłady problemów algorytmicznych

Sortowanie danych:

DANE: tablica (a_1, a_2, \dots, a_n) o n elementach typu porządkowego,

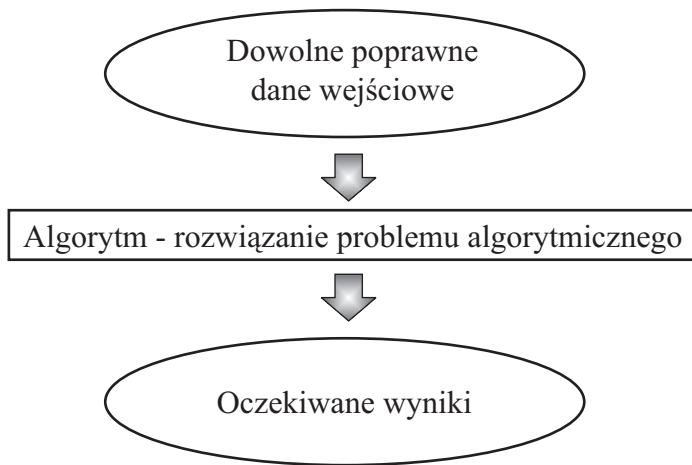
SZUKANE: tablica o tych samych elementach ale uporządkowana np.:
niemalejąco.

Problem komiwojażera TSP (ang. Travelling Salesman Problem):

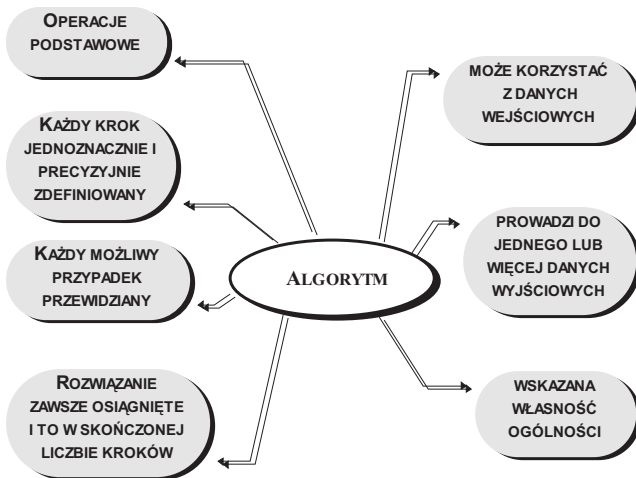
DANE: n miast, odległości pomiędzy miastami $(d_{ij} | i, j = 1, 2, \dots, n)$,

SZUKANE: trasa komiwojażera przez wszystkie miasta (ale tylko jedna wizyta w każdym mieście - permutacja miast) o najmniejszej sumie odległości.

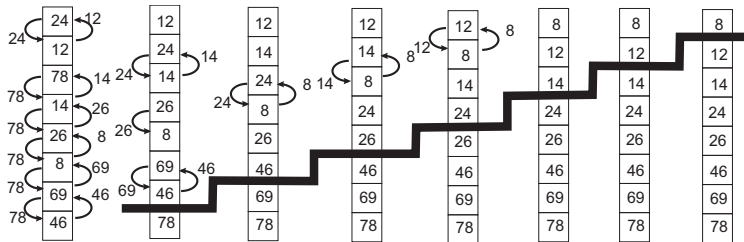
Czym jest algorytm?



Własności algorytmu



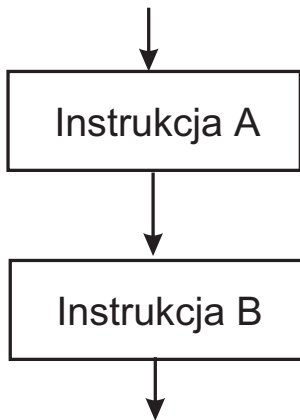
Przykład algorytmu - sortowanie bąbelkowe



Struktury sterujące

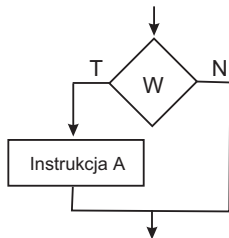
- bezpośrednie następstwo;
- wybór warunkowy;
- iteracja ograniczona;
- iteracja warunkowa (nieograniczona);
- *instrukcja skoku*.

Bezpośrednie następstwo

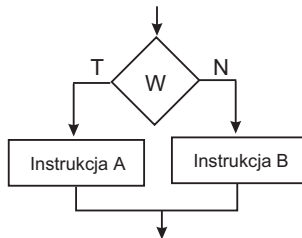


Wykonaj A,
a potem B

Wybór warunkowy



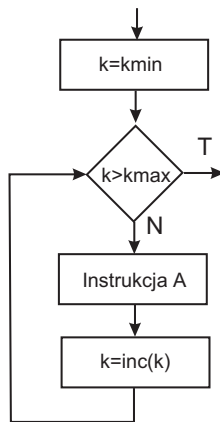
Jeżeli spełniony
jest warunek W
to wykonaj A



Jeżeli spełniony
jest warunek W
to wykonaj A,
w przeciwnym
wypadku
wykonaj B

Iteracja ograniczona

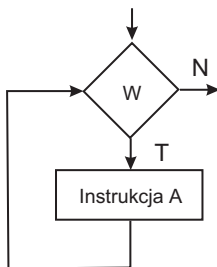
Zazwyczaj reprezentowana przez pętlę typu **for**:



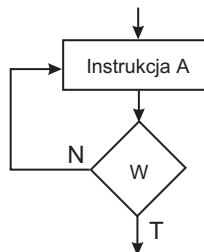
Dla k od k_{\min}
do k_{\max}
wykonuj A

Iteracja warunkowa (nieograniczona)

Zazwyczaj reprezentowana przez pętle typu **while** lub **repeat/until**:



Dopóki spełniony
jest warunek W
wykonuj A
(*while W do A*)

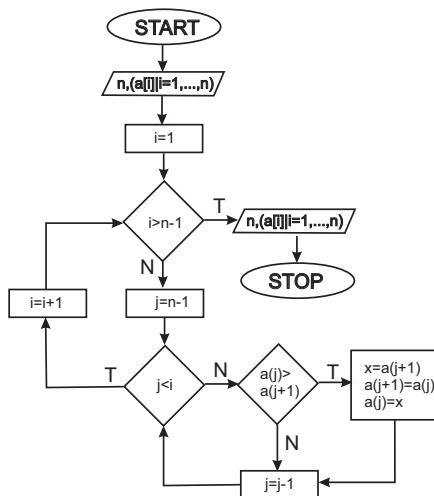


Wykonuj A dopóki
nie zostanie spełniony
warunek W
(*repeat A until W*)

Algorytm sortowania bąbelkowego

- 1 dla i zmieniającego się od 1 do $n-1$ wykonuj co następuje:
 - 1.1 dla j zmieniającego się od $n-1$ w dół do i wykonuj co następuje:
 - 1.1.1 jeżeli $a(j) > a(j+1)$ wówczas:
 - 1.1.1.1 x staje się $a(j)$
 - 1.1.1.2 $a(j)$ staje się $a(j+1)$
 - 1.1.1.3 $a(j+1)$ staje się x

Schemat blokowy sortowania bąbelkowego



Przykłady metod układania algorytmów

- 1 **metoda „dziel i zwyciężaj”**: problem rozmiaru n dzielimy na podproblemy mniejszych rozmiarów, w taki sposób, że z ich rozwiązań wynika rozwiązanie zasadnicze;
- 2 **programowanie dynamiczne**: poprawia powyższą metodę w sytuacjach, kiedy wymaga wielokrotnego liczenia rozwiązań tych samych podproblemów;
- 3 **metoda zachłanna**: w przypadku poszukiwania „najlepszego” rozwiązania spośród znacznej liczby (wykładniczo rosnącej z rozmiarem problemu) możliwych realizacji, staramy się uporządkować proces poszukiwania tak, aby w sensownym czasie mieć szansę znalezienia możliwie najlepszego rozwiązania (niekoniecznie optymalnego).

Przykład metody „dziel i zwyciężaj”: Binary Search

1	Acki	
2	Backi	Dacki > Backi
3	Cacki	
4	Dacki	Dacki = Dacki
5	Ecki	Dacki < Eeck
6	Fecki	
7	Gecki	
8	Hecki	
9	Icki	
10	Kicki	Dacki < Kicki
11	Licki	
12	Micki	
13	Nicki	
14	Ocki	
15	Pocki	
16	Rocki	
17	Socki	
18	Tocki	
19	Ucki	
20	Wucki	

Programowanie dynamiczne - dwumian Newtona

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \quad \binom{n}{0} = 1, \quad \binom{n}{n} = 1$$

dziel i rządź

$\binom{5}{3}$	$\binom{4}{2}$	$\binom{3}{1}$	$\binom{2}{0}$	
			$\binom{2}{1}$	$\binom{1}{0}$
				$\binom{1}{1}$
		$\binom{3}{2}$	$\binom{2}{1}$	$\binom{1}{0}$
				$\binom{1}{1}$
			$\binom{2}{2}$	
	$\binom{4}{3}$	$\binom{3}{2}$	$\binom{2}{1}$	$\binom{1}{0}$
				$\binom{1}{1}$
			$\binom{2}{2}$	
		$\binom{3}{3}$		

programowanie dynamiczne

$\binom{1}{0}$	$\binom{2}{0}$	$\binom{3}{0}$	$\binom{4}{0}$	$\binom{5}{0}$
$\binom{1}{1}$	$\binom{2}{1}$	$\binom{3}{1}$	$\binom{4}{1}$	$\binom{5}{1}$
	$\binom{2}{2}$	$\binom{3}{2}$	$\binom{4}{2}$	$\binom{5}{2}$
		$\binom{3}{3}$	$\binom{4}{3}$	$\binom{5}{3}$

Metoda zachłanna - problem plecakowy

Problem: Dany jest zbiór n elementów $S = (s_i, c_i, w_i)_{i=1}^n$, gdzie każdy element s_i posiada swoją wartość c_i i wagę w_i . Należy zapakować plecak o maksymalnej nośności W (tzn. powyżej tej wagi podrze się plecak) elementami ze zbioru S , tak aby wartość zawartości plecaka była jak największa

Możliwe strategie:

- przegląd wszystkich możliwych realizacji (jest ich 2^n);
- najpierw najwartościowsze;
- najpierw najlżejsze;
- najpierw o najwyższym stosunku $\frac{c_i}{w_i}$

A w następnym tygodniu między innymi

Tematy jakie zostaną poruszone na następnym wykładzie:

- 1 sortowanie wewnętrzne – sformułowanie problemu,
- 2 sortowanie bąbelkowe,
- 3 sortowanie przez wybieranie,
- 4 sortowanie przez wstawianie i przez wstawianie połówkowe,
- 5 sortowanie szybkie,
- 6 sortowanie stogowe.

Dziękuję za uwagę!!!