

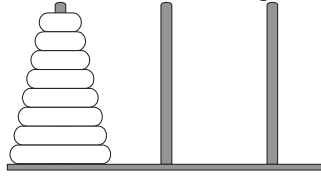
Rekurencja

- zdolność podprogramu (procedury) do wywoływania samego (samej) siebie

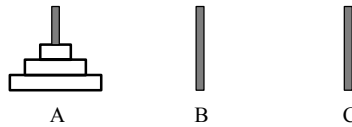
Wieża Hanoi

dane wejściowe - trzy kołki i N krążków o różniących się średnicach

wynik - sekwencja ruchów przenosząca krążki z kołka na kołek zgodnie z zadanymi regułami



Zacznijmy od przypadku szczególnego dla $N = 3$:

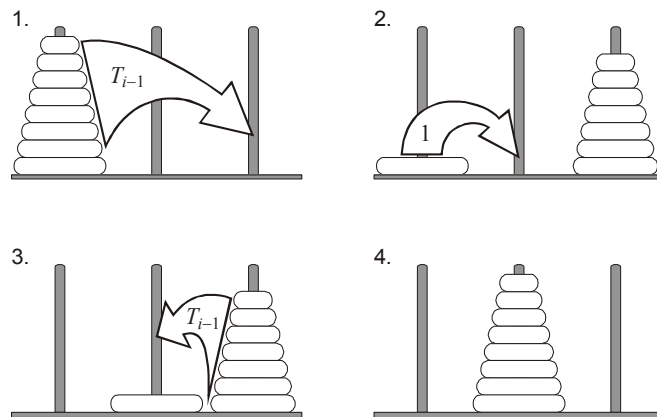


$X \rightarrow Y$ oznacza przeniesienie szczytowego krążka z kołka X na Y

Rozwiązanie dla $N = 3$:

$A \rightarrow B$; $A \rightarrow C$; $B \rightarrow C$; $A \rightarrow B$; $C \rightarrow A$; $C \rightarrow B$; $A \rightarrow B$

Idea rozwiązania rekurencyjnego dla N krążków:

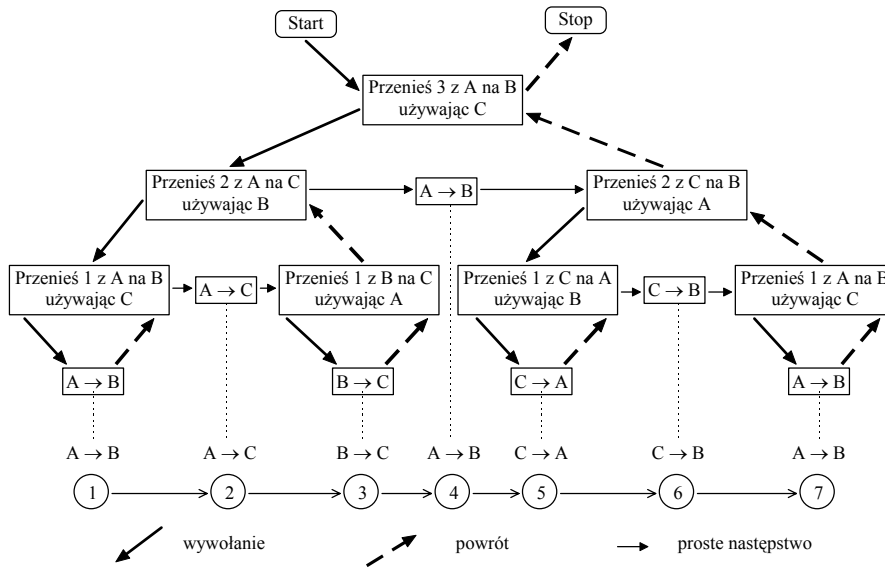


procedura przenieś N z X na Y używając Z :

1. jeśli $N = 1$, to wypisz „ $X \rightarrow Y$ ”;
2. w przeciwnym razie (tj. jeśli $N > 1$) wykonaj co następuje:
 - 2.1. wywołaj *przenieś $N - 1$ z X na Z używając Y* ;
 - 2.2. wypisz „ $X \rightarrow Y$ ”;
 - 2.3. wywołaj *przenieś $N - 1$ z Z na Y używając X* ;
3. wróć do poziomu wywołania.

Rozwiązanie dla $N = 3$:

wywołaj *przenieś 3 z A na B używając C* ;



Minimalny zbiór struktur sterujących:

- nastęstwo (*a-potem*)
- wybór warunkowy (*jeśli-to*)
- jeden z rodzajów iteracji nieograniczonej (np. *dopóki-wykonuj*)

Instrukcja skoku może być usunięta z każdego algorytmu!

Podprogramy rekurencyjne są przy pewnych założeniach silniejsze niż iteracje, ale z większości algorytmów rekurencję można „usunąć” i zastąpić iteracjami.

Typy danych:

- liczby (całkowite, dziesiętne, dwójkowe itp.)
- znaki (układy liter z różnych alfabetów)
- wskaźniki (dane tego typu należy traktować jako adresy obiektów w pamięci operacyjnej - wymagają one specjalnego traktowania)

Dane liczbowe:

liczba dwójkowa: $11001101 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

liczba dziesiętna: $205 = 2 \cdot 10^2 + 0 \cdot 10^1 + 5 \cdot 10^0$

liczba heksadecymalna: $1CD = 1 \cdot 16^2 + 12 \cdot 16^1 + 13 \cdot 16^0$

„A” ~ $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \sim 2^6 + 2^0 = 65$

„B” ~ $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \sim 2^6 + 2^1 = 66$

Dane znakowe:

Tablica kodów ASCII (z ang. *American Standard Code for Information Interchange*)

Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak
0		16		32	SP	48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	\$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	'	55	7	71	G	87	W	103	g	119	w
8		24		40	(56	8	72	H	88	X	104	h	120	x
9		25		41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12		28		44	,	60	<	76	L	92	\	108	l	124	
13	CR	29		45	-	61	=	77	M	93]	109	m	125	}
14		30		46	.	62	>	78	N	94	^	110	n	126	~
15		31		47	/	63	?	79	O	95	_	111	o	127	

Struktury danych:

organizują obiekty w pamięci służące do przechowywania elementów zbioru danych, którymi manipulują algorytmy:

- struktury statyczne
- struktury dynamiczne

ZMIENNE (podstawowe obiekty w pamięci)

- mają nadaną nazwę i zdolność przechowywania pojedynczego elementu zbioru danych
- są obiektami wykreowanymi w **pamięci** (operacyjnej) dla potrzeb algorytmu

$X \leftarrow 0$ oznacza nadanie zmiennej X wartości 0
(tj. określenie zawartości pewnego obiektu w pamięci)

$X \leftarrow X + 1$ oznacza odczytanie aktualnej wartości zmiennej X , dodanie do niej jednośc i zapisanie wyniku w tym samym obiekcie (miejscu) w pamięci, które symbolizuje nazwa zmiennej X

STRUKTURY STATYCZNE

Tablice jednowymiarowe (wektory)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które ponumerowano liczbami naturalnymi – każda z nich ma przypisany na stałe tzw. indeks,
- mogą przechowywać nie większą od ich długości liczbę elementów zbioru danych jednakowego typu.

Tablica T :

T							- nazwa
15	11	24	36	17	15	51	- wartości
1	2	3	4	5	6	7	- indeksy

W zapisie symbolicznym $T(6)$ oznacza 6 zmienną w tablicy T

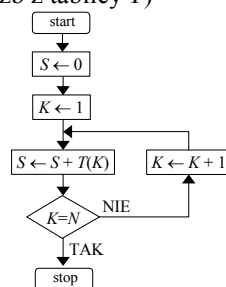
Indeks może być określony przez bezpośrednie podanie wartości w odwołaniu do elementu tablicy, np. $T(6)$, lub użycie nazwy zmiennej o typie zgodnym z indeksem, np. $T(X)$. Zmienną X nazywamy wtedy zmienną indeksową i wskazanie elementu tablicy wymaga odczytania jej aktualnej wartości.

Przykłady zastosowania statycznych struktur danych

Algorytm sumowania N liczb zapamiętanych w tablicy T

1. $S \leftarrow 0$ (ustalenie początkowej wartości sumy);
2. $K \leftarrow 1$ (ustalenie początkowej wartości zmiennej indeksowej);
3. wykonaj co następuje N razy:
 - 3.1. $S \leftarrow S + T(K)$;
 - 3.2. $K \leftarrow K + 1$.

(po zakończeniu zmienna S ma wartość sumy liczb z tablicy T)



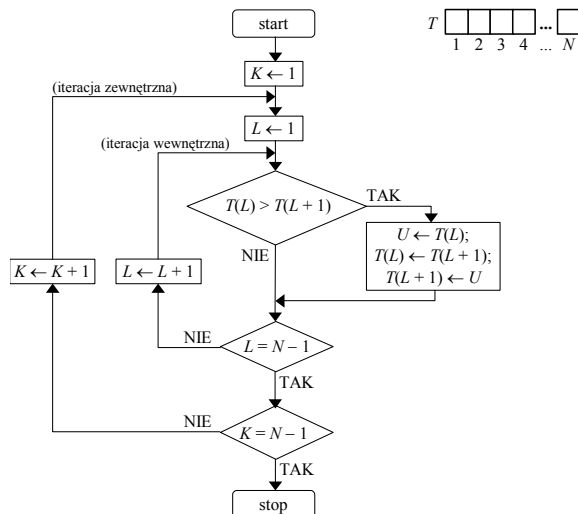
W algorytmie sortowania bąbelkowego dla listy N liczb rzeczywistych (porównania decydujące o kolejności elementów wykonywane będą w oparciu o relację $<$) możemy użyć:

tablicy T o długości co najmniej N , zmiennych K i L do sterowania iteracjami wewnętrzną i zewnętrzną oraz pomocniczej zmiennej U .

Zapis algorytmu:

1. $K \leftarrow 1$ (ustalenie początkowej wartości zmiennej indeksowej);
2. wykonaj co następuje $N - 1$ razy:
 - 2.1. $L \leftarrow 1$;
 - 2.2. wykonaj co następuje $N - 1$ razy:
 - 2.2.1. jeśli $T(L + 1) < T(L)$ to:
 $U \leftarrow T(L)$; $T(L) \leftarrow T(L + 1)$; $T(L + 1) \leftarrow U$;
 - 2.2.2. $L \leftarrow L + 1$;
 - 2.3. $K \leftarrow K + 1$.

Schemat blokowy:



Tablice dwu – i więcej wymiarowe (macierze)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które oznaczono dwoma lub więcej indeksami,
- mogą przechowywać nie większą od ich rozmiaru liczbę elementów zbioru danych jednakowego typu.

Tablica W :

nazwa - W		- indeksy kolumn						
		1	2	3	4	5	6	7
1		15	11	24	36	17	15	51
2		14	32	28	26	19	20	43
3		11	16	13	31	10	15	41
		- wartości						
		indeksy wierszy						

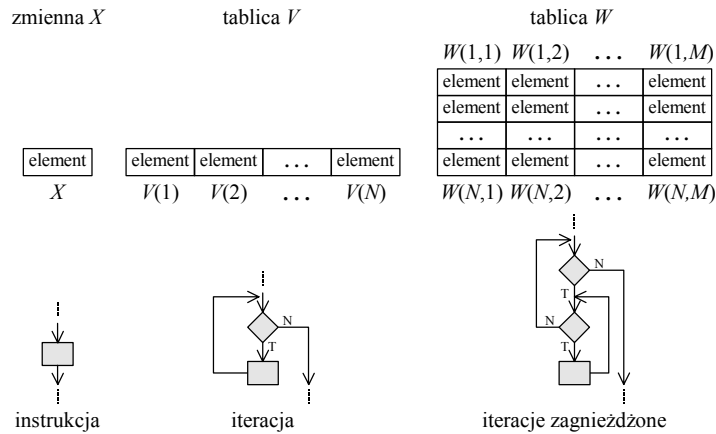
W zapisie symbolicznym $W(3, 5)$ oznacza zmienną w tablicy W położoną umownie na przecięciu 3. wiersza i 5. kolumny.

Algorytm sumowania $N \times M$ liczb zapamiętanych w tablicy W

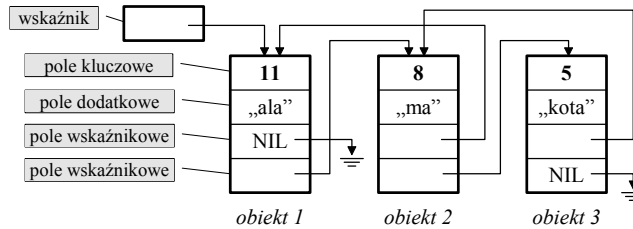
1. $S \leftarrow 0$ (ustalenie początkowej wartości sumy);
2. $K \leftarrow 1$ (ustalenie początkowej wartości 1. zmiennej indeksowej);
3. wykonaj co następuje N razy:
 - 3.1. $L \leftarrow 1$ (ustalenie początkowej wartości 2. zm. indeksowej);
 - 3.2. wykonaj co następuje M razy:
 - 3.2.1. $S \leftarrow S + W(K, L)$;
 - 3.2.2. $L \leftarrow L + 1$.
 - 3.3. $K \leftarrow K + 1$.

(po zakończeniu zmienna S ma wartość sumy liczb z tablicy W)

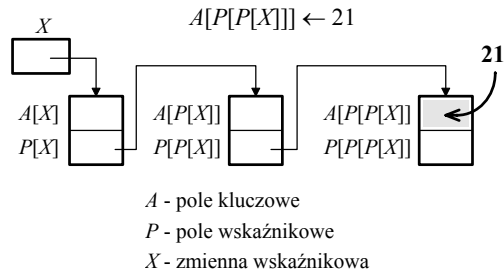
Stacyczne struktury danych i związane z nimi struktury sterujące



STRUKTURY DYNAMICZNE (WSKAŹNIKOWE)



Odwolania do pól obiektów za pomocą wskaźników:

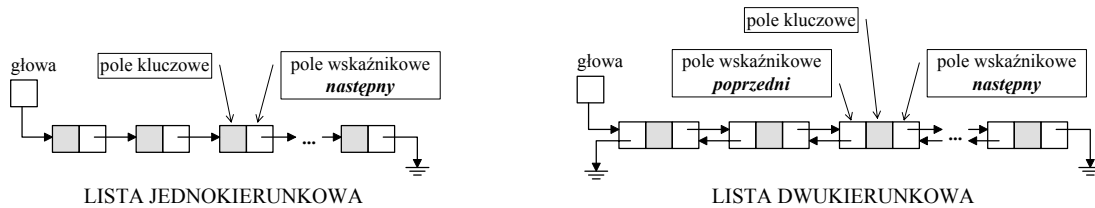


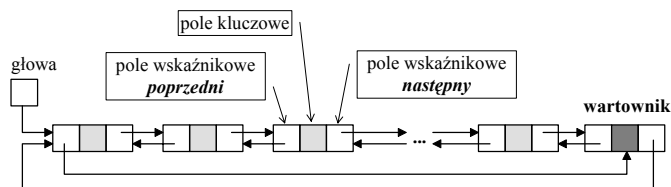
Podstawowe operacje modyfikujące struktury dynamiczne:

- DODAJ (ang. *INSERT*), czyli dołącz we wskazanym miejscu nowy obiekt do struktury (wcześniej trzeba zainicjować ten nowy obiekt w pamięci)
- USUŃ (ang. *DELETE*), czyli odłącz od struktury wskazany obiekt (i zwolnij zajmowane przez niego miejsce w pamięci)

LISTY (dynamiczne struktury liniowe)

- listy z dowiązaniem jednokierunkowym
- listy z dowiązaniem dwukierunkowym
- listy z wartownikiem



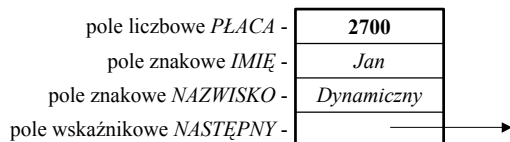


LISTA Z WARTOWNIKIEM

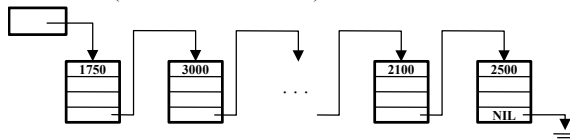
Przykład zastosowania dynamicznej struktury danych (listy)

W algorytmie sumowania płac pracowników możemy użyć:

- rekordu (zespołu zmiennych) do przechowywania danych pojedynczego pracownika:



- listy do przechowywania danych o wszystkich pracownikach firmy
PIERWSZY (zmienna wskaźnikowa)



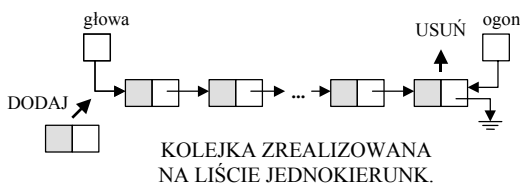
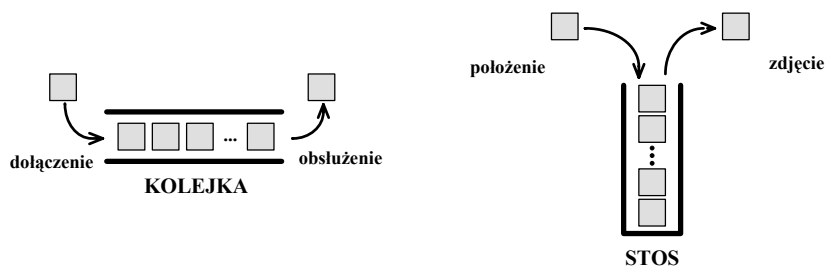
- zmiennej wskaźnikowej *X* do wskazywania rekordu z danymi kolejnego pracownika
- zmiennej *S* do kumulowania wyniku.

Zapis algorytmu:

1. $S \leftarrow 0$;
2. $X \leftarrow \text{PIERWSZY}$;
3. dopóki $X \neq \text{NIL}$, wykonuj co następuje:
 - 3.1. $S \leftarrow S + \text{PLACA}[X]$;
 - 3.2. $X \leftarrow \text{NASTĘPNY}[X]$;
4. podaj wartość *S* jako wynik.

KOLEJKI I STOSY (struktury dynamiczne o ograniczonych możliwościach modyfikacji)

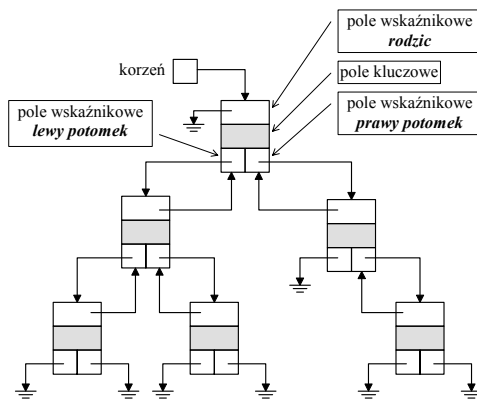
- kolejka - struktura FIFO (First-In-First-Out)
- stos - struktura LIFO (Last-In-First-Out)



KOLEJKA ZREALIZOWANA NA LIŚCIE JEDNOKIERUNK.

DRZEWA (rozgałęzione struktury dynamiczne)

- drzewa binarne
- drzewa o dowolnym stopniu rozgałęzień
- drzewa BST (ang. Binary Search Tree) - poszukiwań binarnych

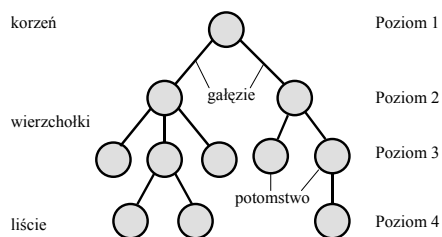


DRZEWO BINARNE

Drzewo binarne - drzewo, w którym rząd wyjściowy wierzchołków ograniczony jest przez 2

Drzewo pełne - drzewo, w którym wszystkie wierzchołki poza liśćmi mają jednakową liczbę potomków i wszystkie liście są na tym samym poziomie

Podstawowe elementy drzew:



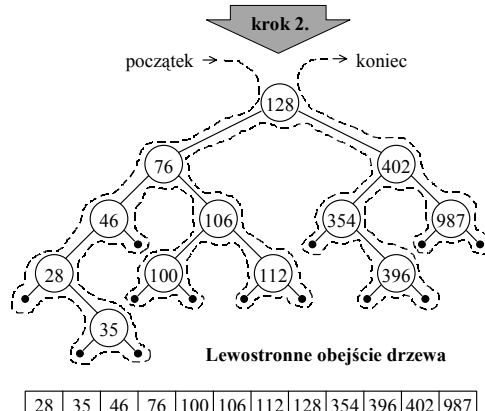
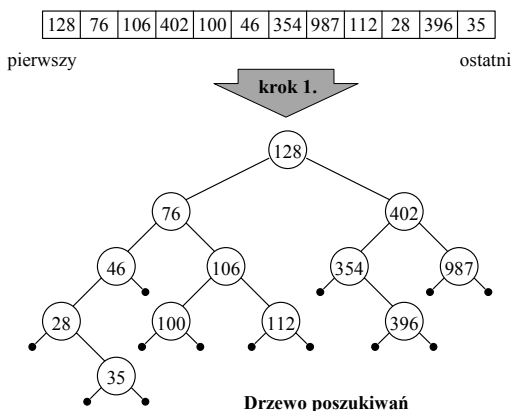
korzeń – wierzchołek bez rodziców;

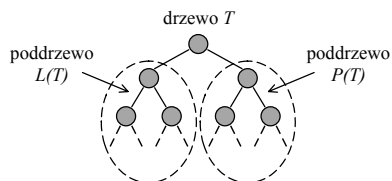
liść – wierzchołek bez potomstwa

Sortowanie drzewiaste

Dwa etapy algorytmu sortowania drzewiastego:

1. przekształć nieuporządkowaną listę wejściową w binarne drzewo poszukiwań *T*;
2. obejdź drzewo *T* w kolejności „najpierw w lewo” i wypisz elementy danych umieszczone w wierzchołkach przy ich powtórnych odwiedzinach.





procedura rekurencyjna realizująca 2 krok sortowania drzewiastego:

procedura **obejdź** T :

1. jeśli T jest puste, to wróć;
2. w przeciwnym razie wykonaj co następuje:
 - 2.1. wywołaj **obejdź** $L(T)$;
 - 2.2. wypisz element danych umieszczony w korzeniu T ;
 - 2.3. wywołaj **obejdź** $P(T)$;
3. wróć.

Rekurencja jest strukturą sterującą najczęściej związaną z **drzewem** jako strukturą danych.

Bazy danych

Duże zbiory danych różnych typów uporządkowane według modelu relacyjnego, hierarchicznego lub innego.

Zapytania (kwerendy) służą do wydobycia potrzebnej informacji.

Bazy wiedzy

Duże zbiory zdań orzekających lub implikacji wyrażonych w języku naturalnym.