

INFORMATYKA

DANE

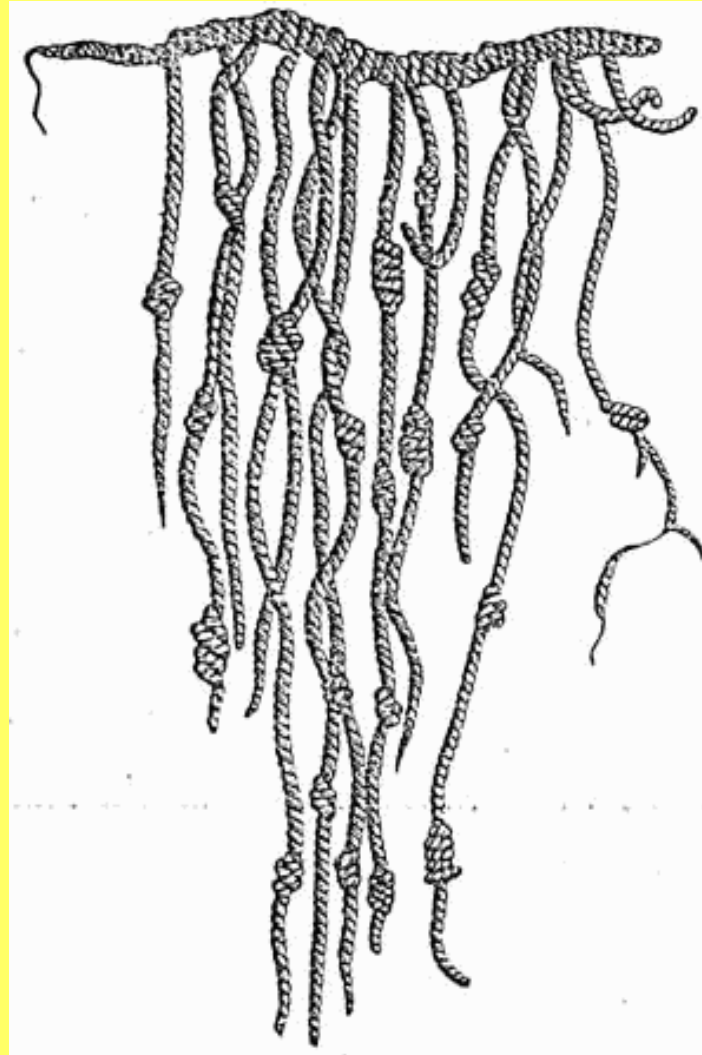
<http://www.infoceram.agh.edu.pl>

DANE

Dane to zbiory liczb, znaków, sygnałów, wykresów, tekstów, itp., które mogą być przetwarzane. Pojęcie danych jest relatywne i istnieje tylko razem z pojęciem przetwarzania danych. Przetwarzanie danych oznacza przekształcanie danych wejściowych w ściśle określony sposób w celu uzyskania wyników w z góry określonej postaci.

Różne dane mogą dostarczać tę samą informację, ale jednocześnie te same dane mogą też dostarczać różnych informacji. Z drugiej strony, np. zbiory liczb czy wyrazów mogą być danymi, o ile znane jest ich znaczenie. W przeciwnym przypadku nie są informacjami.

KIPU – przykład zbioru danych nie będących dla nas informacją



Typy danych

Typ danych to opis rodzaju, struktury i zakresu wartości, jakie mogą przyjmować dane.

Każda stała, zmienna, wyrażenie, funkcja jest daną pewnego typu.

Typ danych określa zbiór wartości

- do którego należy stała
- które może przyjmować zmienna i wyrażenie
- które może zwracać funkcja lub operator

Podział typów danych

- proste
 - arytmetyczne (całkowite, rzeczywiste)
 - znakowe
- złożone
 - tablice
 - struktury

PRZYKŁADOWE TYPY DANYCH

typ całkowity – typ reprezentujący liczby całkowite z określonego zakresu

typ zmiennoprzecinkowy – typ reprezentujący przybliżoną wartość liczby rzeczywistej

typ stałopozycyjny – typ reprezentujący liczbę wymierną o stałym mianowniku.

typ znakowy – typ reprezentujący pojedynczy znak (ASCII lub Unicode)

typ tekstowy – typ reprezentujący cały tekst.

typ wyliczeniowy – typ mogący przyjmować jedną z zadanych symbolicznych wartości, np. (czerwony, zielony, niebieski)

typ tablicowy – ciąg zmiennych danego typu, indeksowanych liczbą naturalną lub dowolnym unikalnym kluczem który może być zarówno liczbą jak i ciągiem znaków

typ logiczny – typ przyjmujący wartości logiczne 1 (*true, t*) lub 0 (*false, nil*)

typ bitowy – typ reprezentujący ciąg bitów

typ zespolony – reprezentujący liczby zespolone.

typ etykietowy – reprezentujący wartości będące etykietami instrukcji, wskazaniem instrukcji.

Typy danych

Typy danych należą do pojęć logicznych, same nie stanowią obiektu, który można wykorzystać w programie.

Reprezentantami typów w programach są zmienne, czyli obiekty posiadające trzy podstawowe atrybuty: symboliczną nazwę (identyfikator), miejsce przechowywania i wartość oraz czasami dodatkowy atrybut, tj. typ.

Zmienne zajmują pewien obszar w pamięci operacyjnej komputera, którego rozmiar i sposób interpretacji określony jest przez typ zmiennej.

Zmienne

Zmienne - służą do przechowywania w pamięci pojedynczych obiektów

Przykład:

Niech **x** będzie zmienną całkowitą.

```
wypisz(X)    //wypisz wartość zmiennej X
X = -2       //przypisz zmiennej X wartość -2
X = X + 5    //zwiększ wartość X o 5
```

STRUKTURA DANYCH

Struktura danych to sposób uporządkowania informacji/danych.

Podstawowe struktury danych:

stos

kolejka

lista

drzewo

graf

tablica

rekord lub struktura

Wybór odpowiedniej dla danego przypadku struktury danych może zmniejszyć złożoność obliczeniową rozwiązywanego problemu, z drugiej jednak strony może prowadzić do wzrostu trudności z implementacją.

STOS

Jest to liniowa struktura danych, w której dane dokładane są na wierzch stosu i z wierzchołka stosu są pobierane (bufor typu **LIFO**, *Last In, First Out; ostatni na wejściu, pierwszy na wyjściu*). Do elementu stosu znajdującego się poniżej wierzchołka jest dostęp tylko po zdjęciu elementów znajdujących się między nim a wierzchołkiem

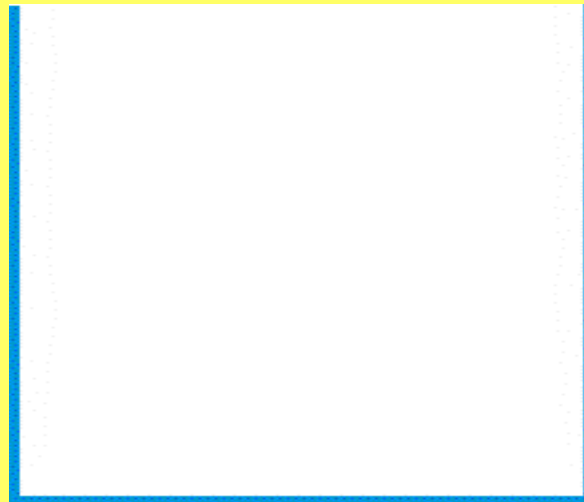
Ideę stosu danych można zilustrować jako stos położonych jeden na drugim talerzy – nowy talerz kładzie się na wierzch stosu i z wierzchu stosu zdejmuje się kolejne egzemplarze.

STOS

Możliwe są następujące operacje na elementach stosu:

- odłożenie na stos jednego elementu
- pobranie ze stosu jednego elementu
- sprawdzenie, czy stos pusty

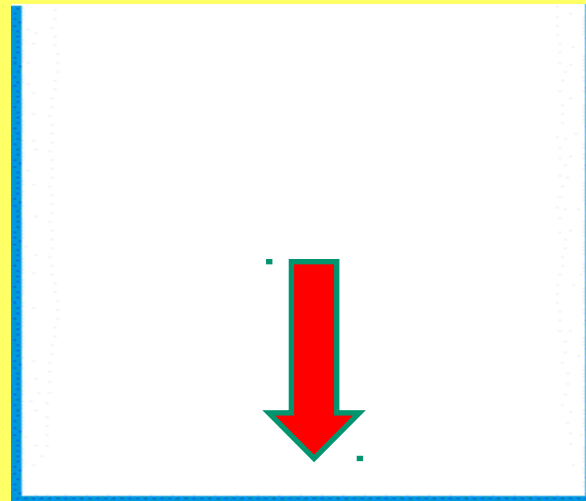
STOS



dno stosu

Stos jest pusty

STOS



dno stosu

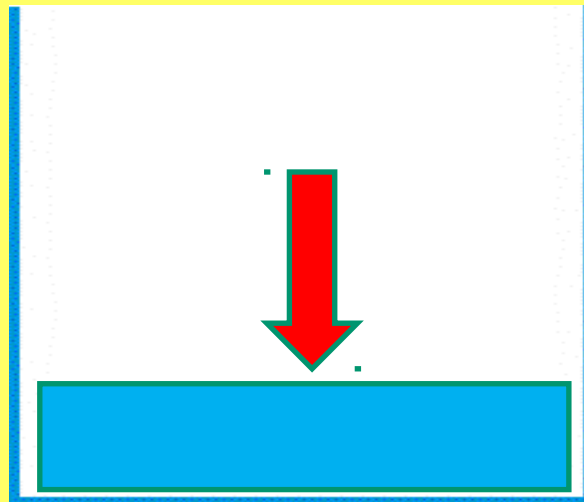
Stos jest pusty

STOS



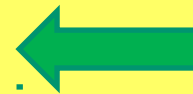
dno stosu

STOS



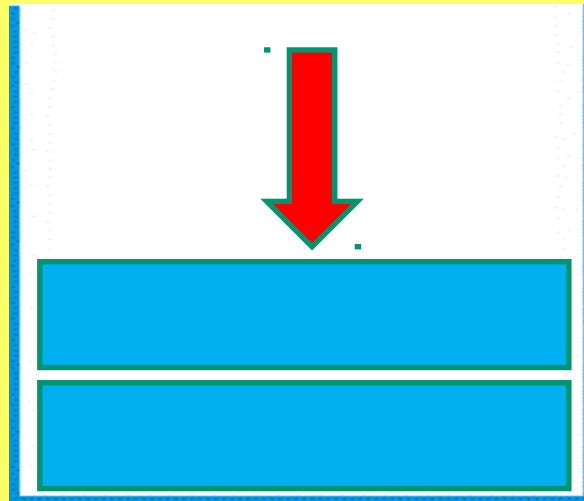
dno stosu

STOS



dno stosu

STOS



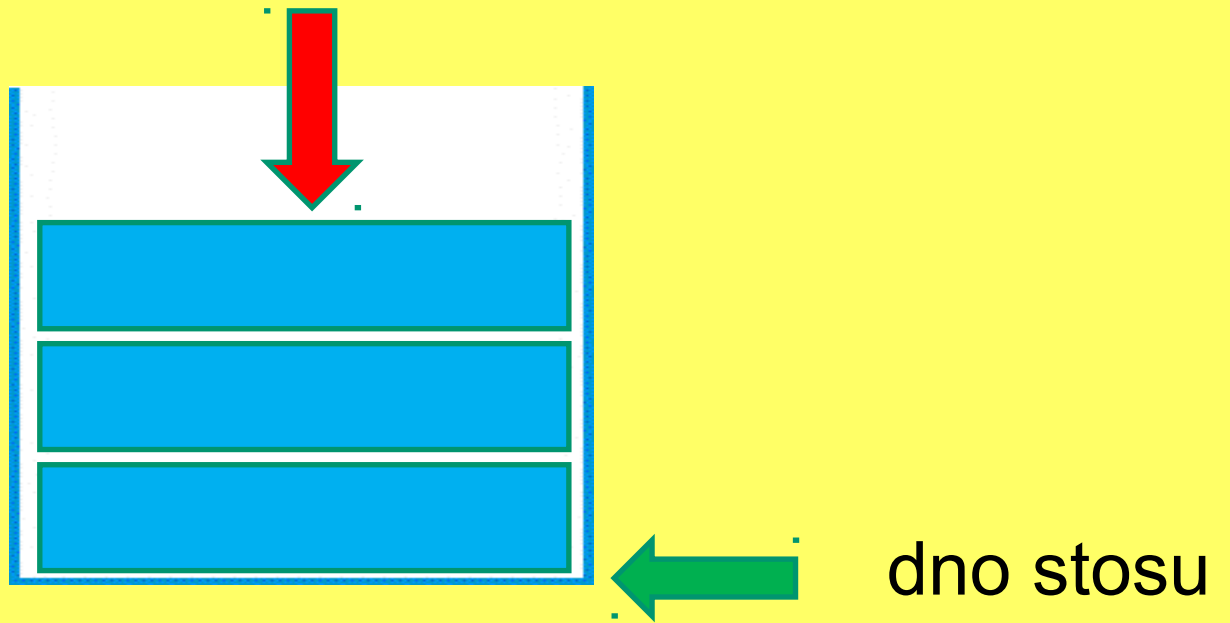
dno stosu

STOS

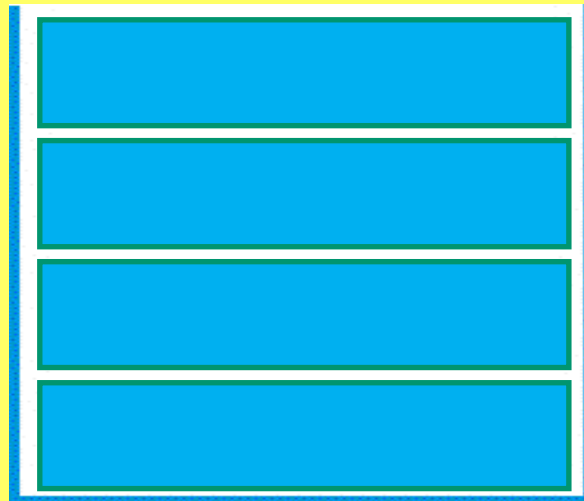


dno stosu

STOS



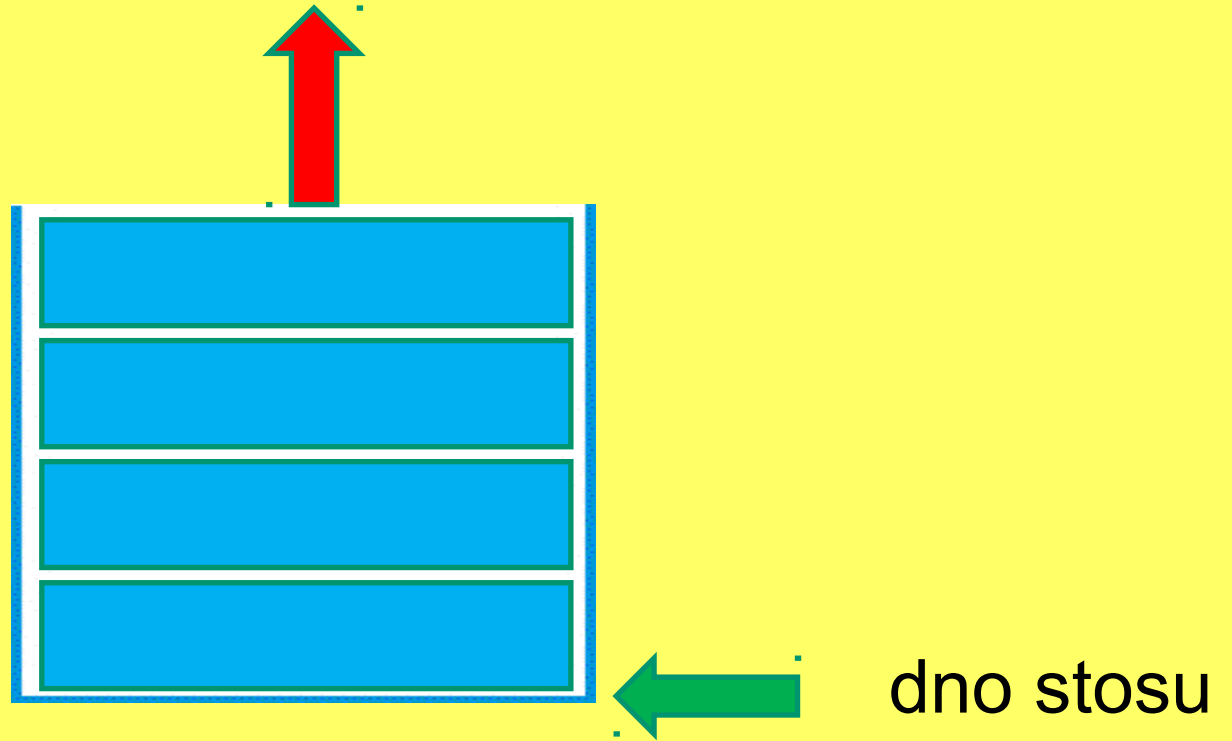
STOS



dno stosu

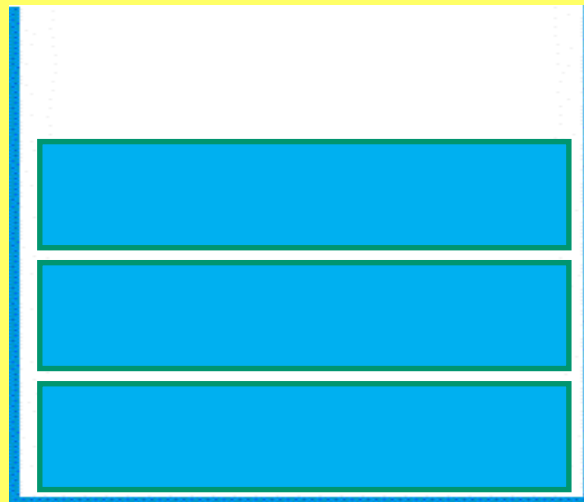
Stos jest pełny

STOS



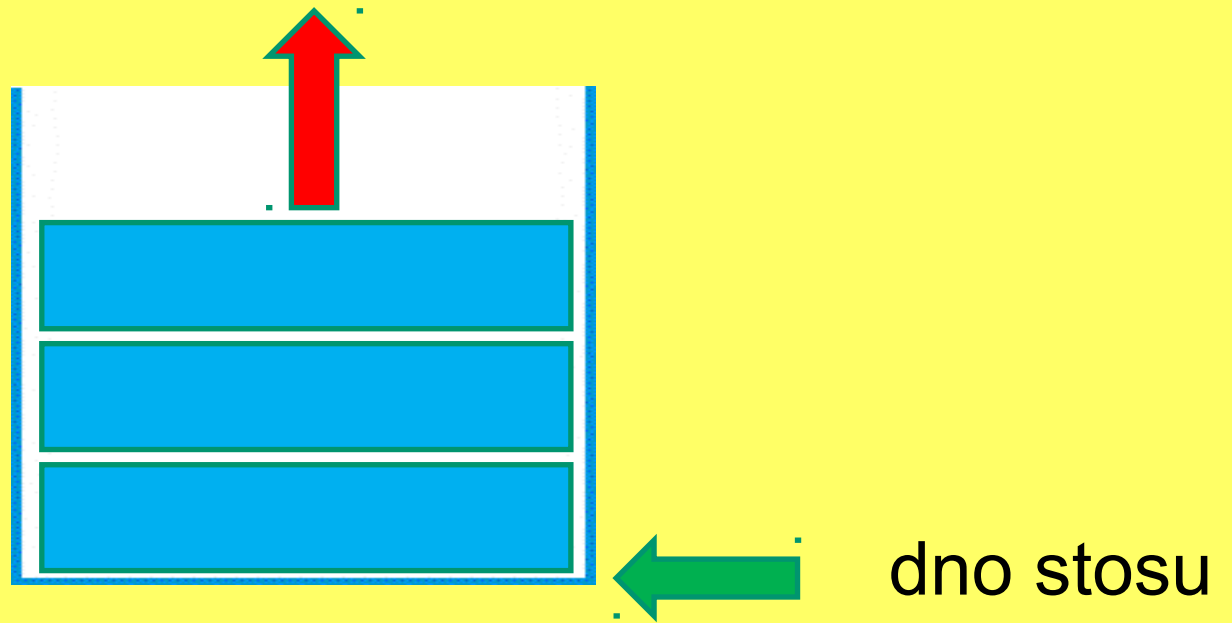
Stos jest pełny

STOS

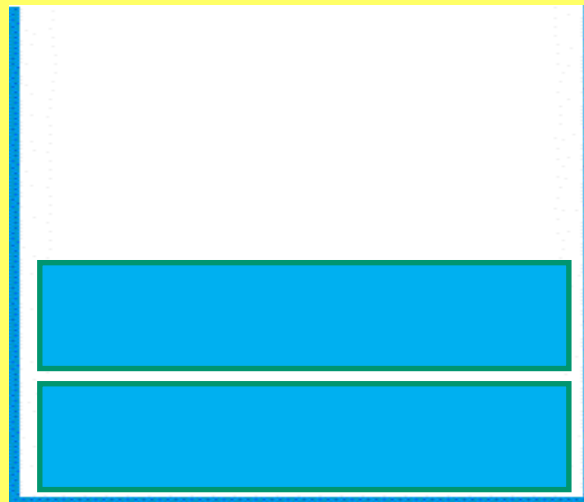


dno stosu

STOS

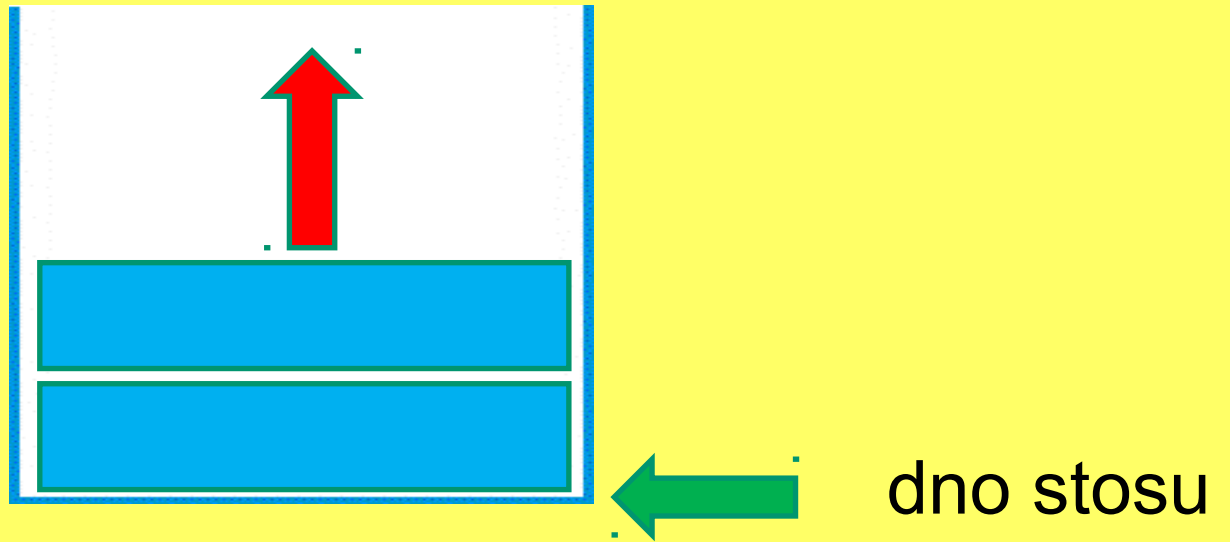


STOS



dno stosu

STOS



STOS

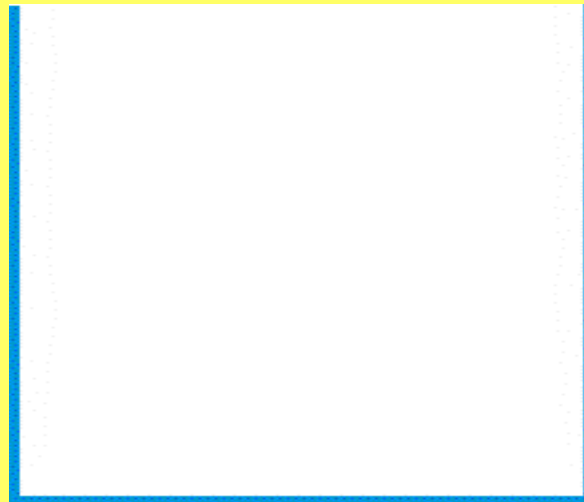


dno stosu

STOS



STOS



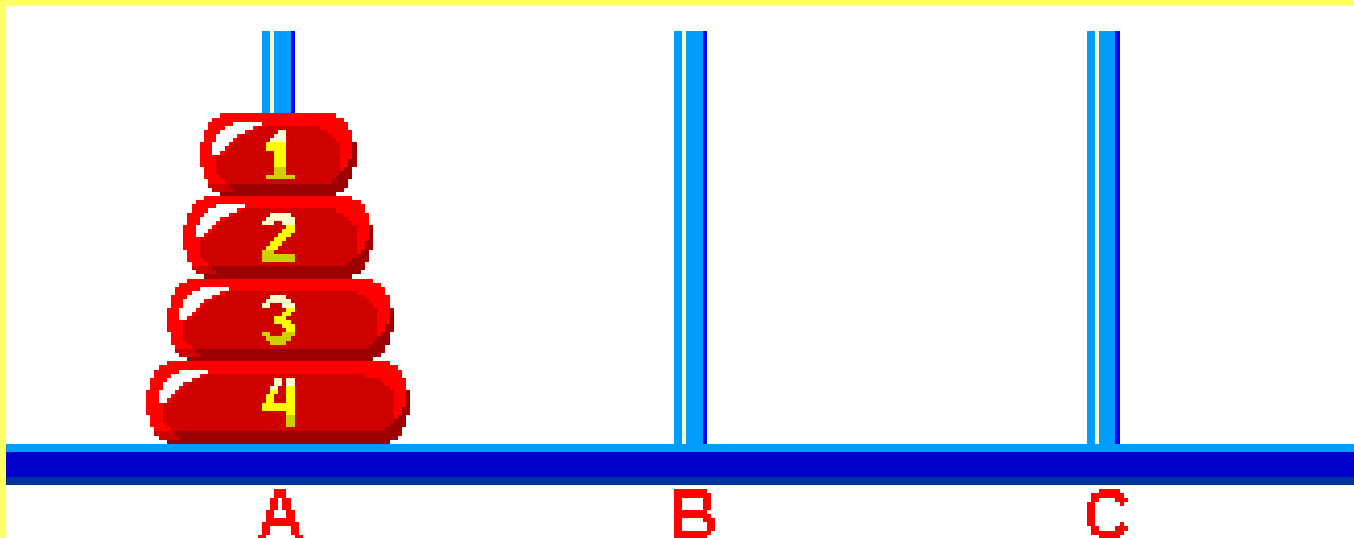
dno stosu

Stos jest pusty

PRZYKŁAD STOSU - WIEŻE HANOI

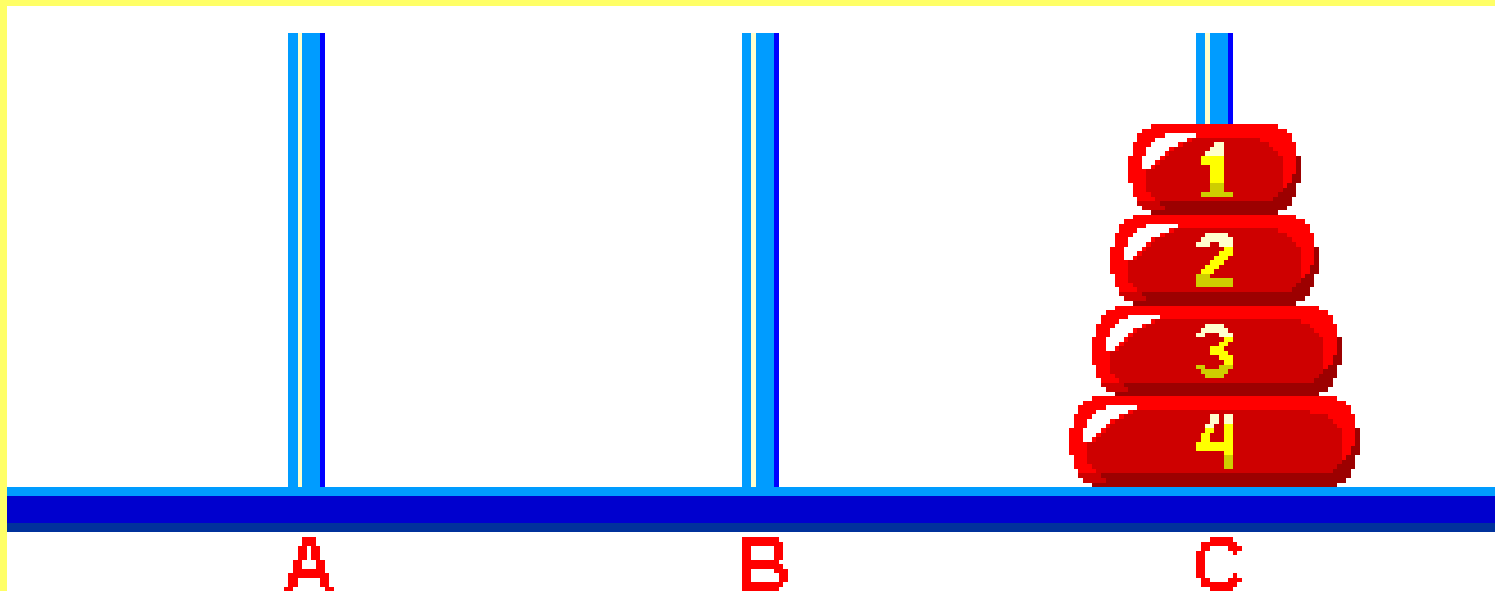
Ułóż stos na trzecim patyku przestrzegając następujących reguł:

- przesuвай jeden krążek z wierzchu stosu,
- nie kładź większego krążka na mniejszym



PRZYKŁAD STOSU - WIEŻE HANOI

Rozwiązaniem zagadki jest ustawienie wieży na słupku 'C' w jak najmniejszej liczbie ruchów, gdzie ruchem jest przeniesienie pojedynczego krążka ze słupka na słupek, z ograniczeniem że nie można kłaść większego krążka na mniejszy.



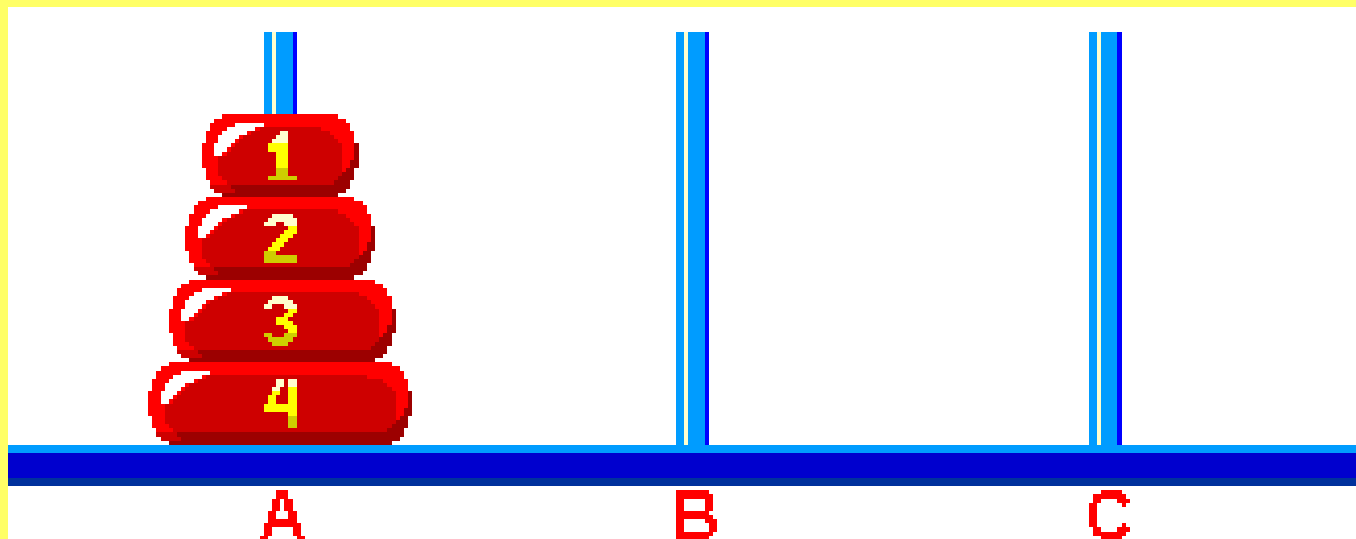
PRZYKŁAD STOSU - WIEŻE HANOI

Rozwiązanie. Niech $A \rightarrow B$ oznacza przeniesienie szczytowego krążka z kołka A na B. Ciąg:

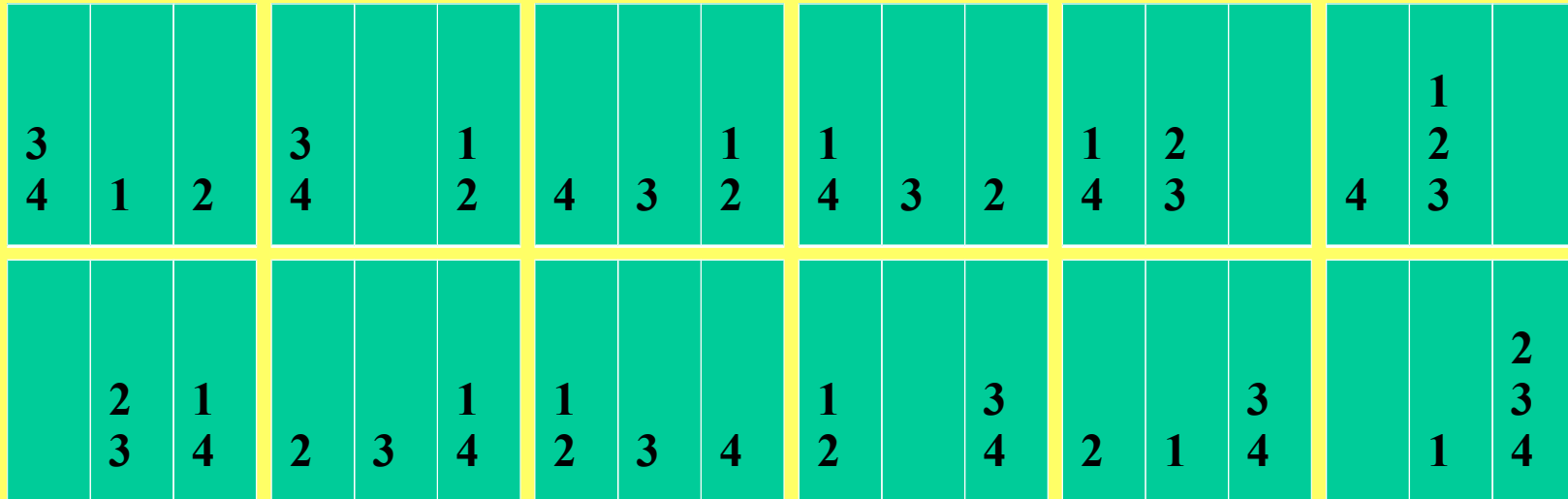
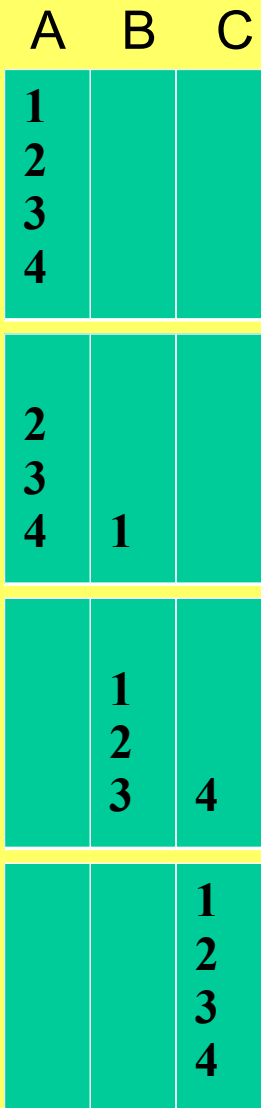
$A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$, $A \rightarrow B$, $C \rightarrow A$, $C \rightarrow B$, $A \rightarrow B$

$A \rightarrow C$, $B \rightarrow C$, $B \rightarrow A$, $C \rightarrow A$, $B \rightarrow C$, $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$

jest rozwiązaniem zadania.



PRZYKŁAD STOSU - WIEŻE HANOI



Rozwiązanie:

$A \rightarrow B, A \rightarrow C, B \rightarrow C, A \rightarrow B, C \rightarrow A, C \rightarrow B, A \rightarrow B$

$A \rightarrow C, B \rightarrow C, B \rightarrow A, C \rightarrow A, B \rightarrow C, A \rightarrow B, A \rightarrow C, B \rightarrow C$

PRZYKŁAD STOSU - WIEŻE HANOI

Rozwiązanie rekurencyjne (B. Wierzba):

1. Zidentyfikuj największy krążek na nie swoim miejscu (=krążek N)
2. If daje się go przenieść na docelowy słupek Then przenieś
Else
3. **Pod zadanie:** ustaw (N-1)-krążkową wieżę na nie-docelowym słupku.
4. Skocz do punktu 1. ...

PRZYKŁAD STOSU - WIEŻE HANOI

```
procedura przenieś(N,A,B,C){ //przenieś N z A na
B                               używając C
    if(N==1)
        wypisz(A," → ", B);
    else{
        przenieś(N-1,A,C,B); //przenieś N-1 z A na C
                               używając B
        wypisz(A," → ",B);
        przenieś(N-1,C,B,A); //przenieś N-1 z C na B
                               używając A
    }
}
```

ODWROTNA NOTACJA POLSKA

Stos znajduje zastosowanie m.in. przy obliczaniu wyrażeń zapisanych za pomocą tzw. odwrotnej notacji polskiej (ONP). ONP jest sposobem zapisu wyrażeń arytmetycznych, w którym znak wykonywanej operacji umieszczony jest po zmiennych, a nie pomiędzy nimi jak w konwencjonalnym zapisie algebraicznym.

Algorytm obliczeń:

Wyzeruj stos.

Dla wszystkich symboli z wyrażenia RPN wykonuj:

jeśli i -ty symbol jest liczbą, to odłóż go na stos,

jeśli i -ty symbol jest operatorem to:

zdejmij ze stosu jeden element (ozn. a),

zdejmij ze stosu kolejny element (ozn. b),

odłóż na stos wartość b operator a .

Zdejmij ze stosu wynik.

ODWROTNA NOTACJA POLSKA

Zapis ten pozwala na całkowitą rezygnację z użycia nawiasów w wyrażeniach, jako że jednoznacznie określa kolejność wykonywanych działań.

ONP bardzo ułatwia wykonywanie na komputerze obliczeń z nawiasami i zachowaniem kolejności działań.

Przykład 1

Zapis konwencjonalny:

$$(3+4)*5$$

zapis w ONP:

$$3 4 + 5 *$$

Przykład 2

Zapis konwencjonalny:

$$((2+3)/5+(10-5)*4)/2$$

zapis w ONP:

$$2 3 + 5 / 10 5 - 4 * + 2 /$$

ODWROTNA NOTACJA POLSKA

Przykład:

Gdy wczytany element jest **liczbą**, to zapisujemy ją na stos.

W przeciwnym wypadku należy wykonać działanie arytmetyczne na 2 ostatnich liczbach na stosie. Wartość wyrażenia znajduje się na stosie.

Wyrażenie $12+2*[(3*4)+(10/5)]$ ONP: **12 2 3 4 * 10 5 / + * +**

Krok	Wejście	Stos
1	12	12
2	2	2 12
3	3	3 2 12
4	4	4 3 2 12
5	*	12 2 12
6	10	10 12 2 12
7	5	5 10 12 2 12
8	/	2 12 2 12
9	+	14 2 12
10	*	28 12
11	+	40

KOLEJKA

Jest to liniowa struktura danych, w której nowe dane dopisywane są na końcu kolejki, a z początku kolejki pobierane są dane do dalszego przetwarzania (bufor typu **FIFO**, *First In, First Out*; *pierwszy na wejściu, pierwszy na wyjściu*). Dostęp do danych jest w takiej kolejności, w jakiej zostały zapisane.

KOLEJKA

Proces dopisywania nowych danych.



początek kolejki

koniec kolejki

KOLEJKA

Proces odczytywania danych.



początek kolejki



koniec kolejki

RODZAJE KOLEJEK

- Kolejka, spotykana jest przede wszystkim w sytuacjach związanych z obsługą zdarzeń.
- Kolejka priorytetowa, w której danym dodatkowo przypisywany jest priorytet, modyfikujący kolejność późniejszego wykonania, co oznacza iż dane pierwsze na wyjściu nie zawsze odczytane będą w pierwszej kolejności, lecz te o największym priorytecie. Kolejkę priorytetową stosuje się m.in. w systemach operacyjnych, gdzie przydzielane są zasoby sprzętowe uruchomionym procesom.
- Kolejka cykliczna, w której pierwszy element uważa się za następny w stosunku do ostatniego.

LISTA

Lista, to liniowa struktura danych tworząca zbiór, z którego w dowolnym miejscu można usunąć element, jak również dołączyć w dowolne miejsce.

Podstawowe rodzaje list:

- jednokierunkowa
- dwukierunkowa

Każdy element listy składa się z przynajmniej dwóch pól: tzw. klucza oraz pola wskazującego na następny element listy. W przypadku list dwukierunkowych każdy element listy zawiera także pole wskazujące na poprzedni element listy.

LISTA

Proces dopisywania nowych danych.

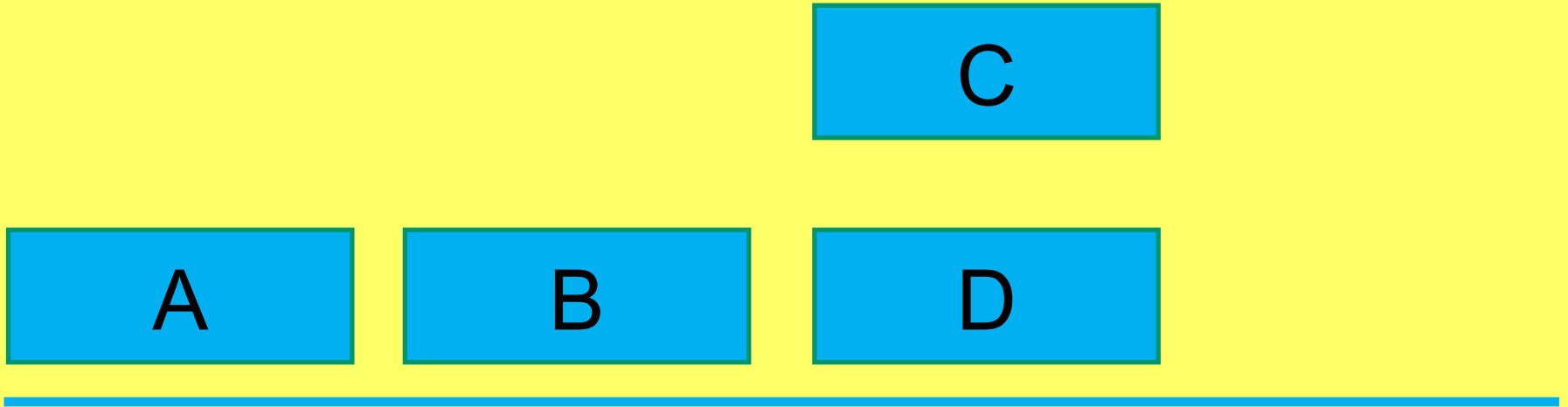
A

B

D

LISTA

Proces dopisywania nowych danych.



LISTA

Proces usuwania danych.

A

B

C

D

DRZEWO

Jest to struktura danych o ułożeniu hierarchicznym. Reprezentując w naturalny sposób hierarchię danych ułatwia i przyspiesza wyszukiwanie, a także pozwala w łatwy sposób operować na posortowanych danych.

Drzewa są stosowane praktycznie w każdej dziedzinie informatyki (bazy danych, grafika komputerowa, przetwarzanie tekstu, telekomunikacja, itp.).

Drzewa składają się z **węzłów** (wierzchołków) oraz łączących je **krawędzi**. Jeden z wierzchołków jest wyróżniony i nazywany **korzeniem drzewa**, natomiast ciąg krawędzi łączących węzły określa się terminem **ścieżka**. Istnieje dokładnie jedna ścieżka łącząca korzeń z wszystkimi pozostałymi węzłami.

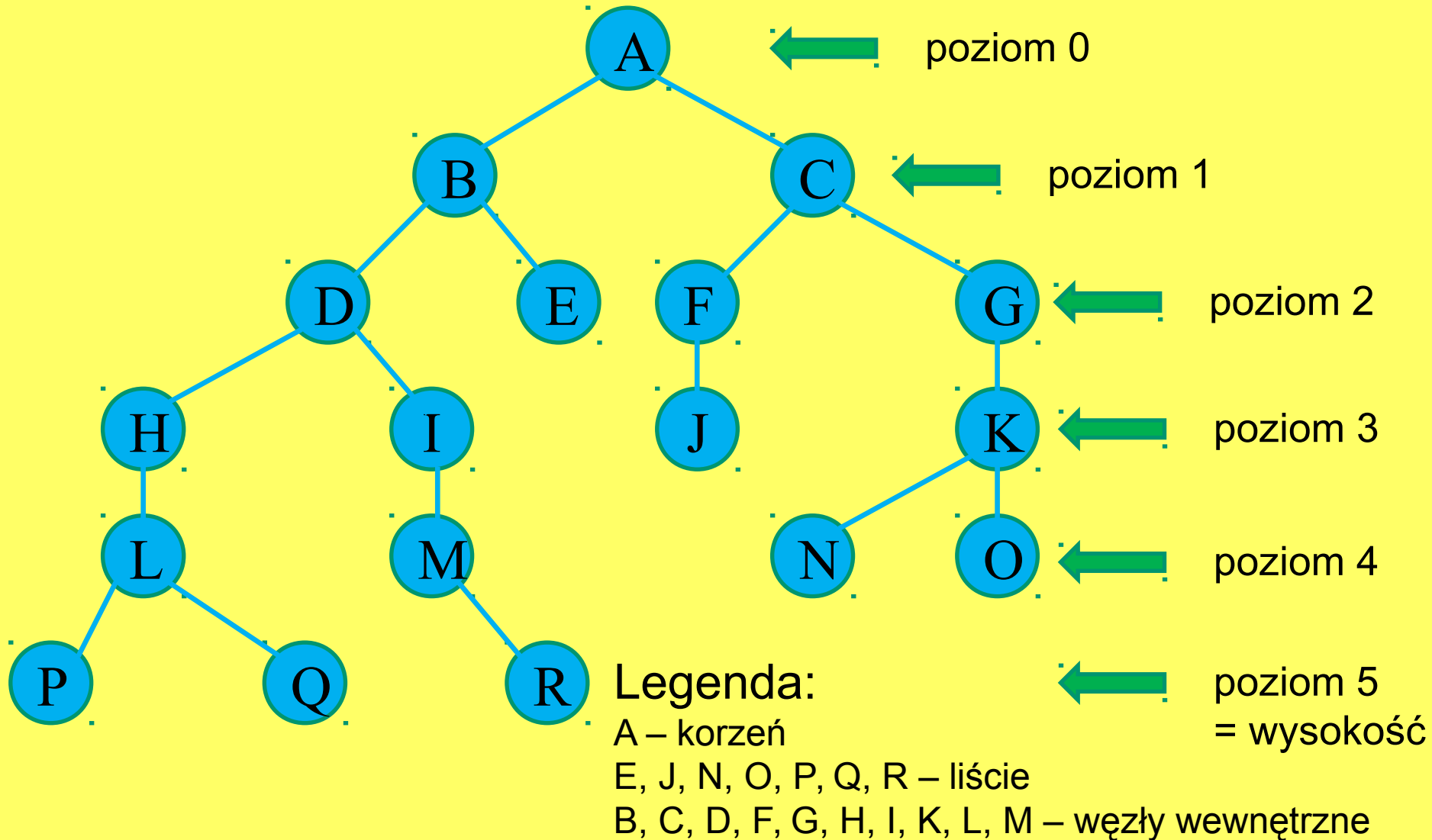
Liczba krawędzi w ścieżce od korzenia do węzła jest nazywana **długością** i określa **poziom** węzła. **Wysokością** drzewa jest największy poziom istniejący w drzewie.

Wszystkie wierzchołki połączone z danym wierzchołkiem, a leżące na następnym poziomie są nazywane jego **dziećmi**. Wierzchołek może mieć dowolną liczbę dzieci, jeśli nie ma ich wcale nazywany jest liściem.

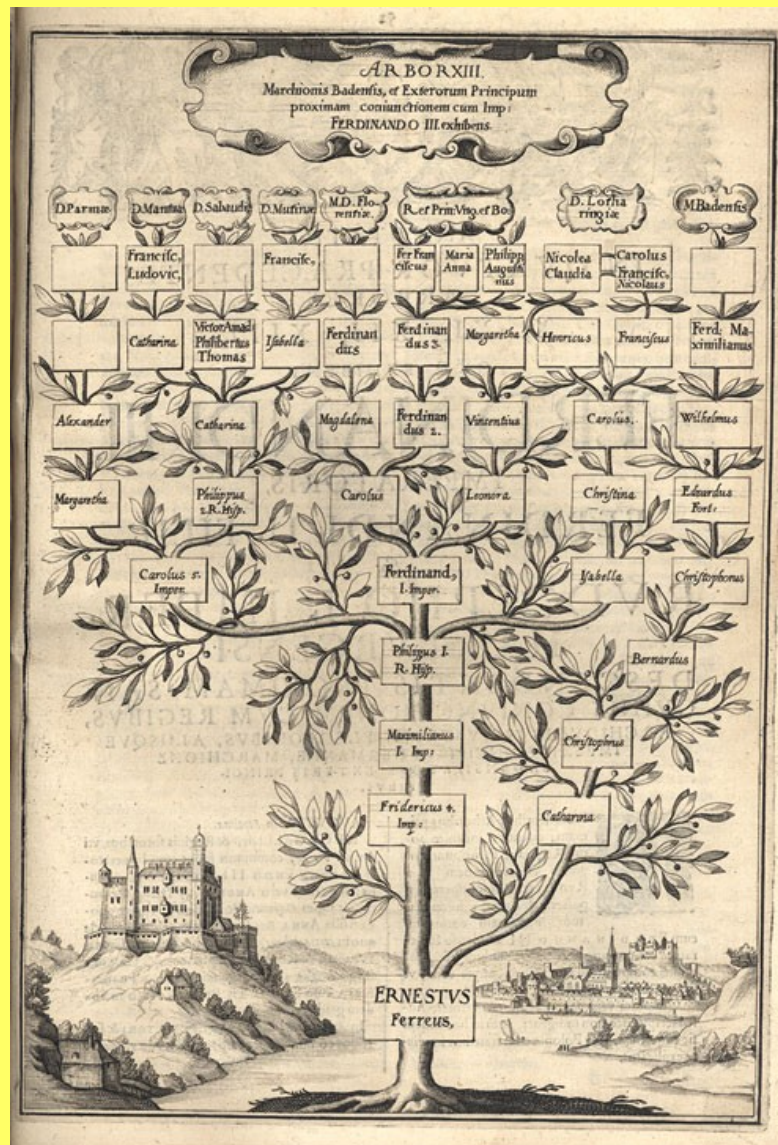
Wierzchołek jest **rodzicem** dla każdego swojego **dziecka**. Każdy węzeł ma dokładnie jednego rodzica, wyjątkiem jest korzeń drzewa, który nie ma rodzica.

Specjalne znaczenie w informatyce mają drzewa binarne, w których liczba dzieci ograniczona jest do dwóch. Drzewa które posiadają więcej niż dwoje dzieci są nazywane **drzewami wyższych rzędów**.

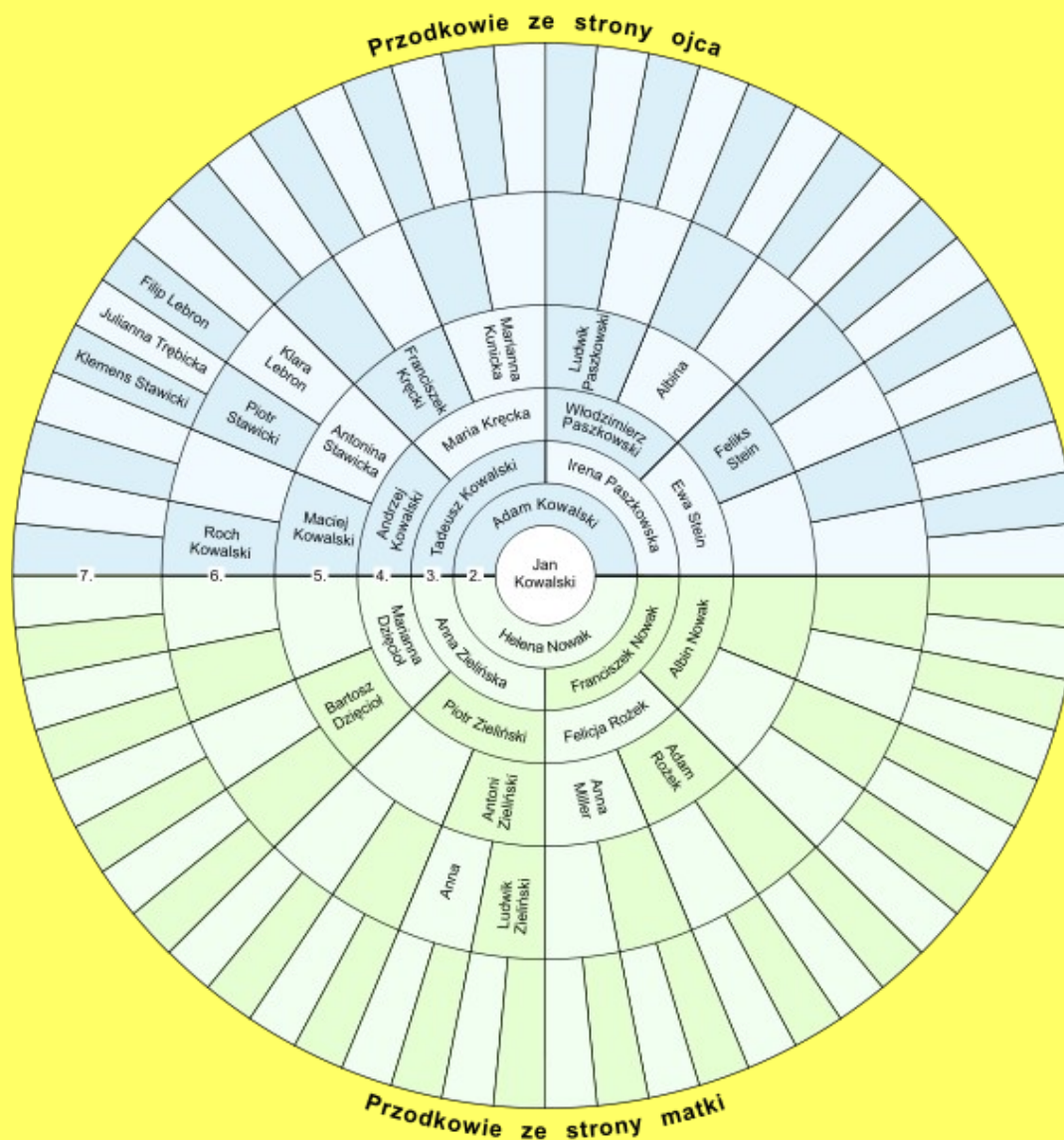
PRZYKŁADOWE DRZEWO BINARNE



DRZEWO GENEALOGICZNE



WYWÓD GENEALOGICZNY – układ kołowy

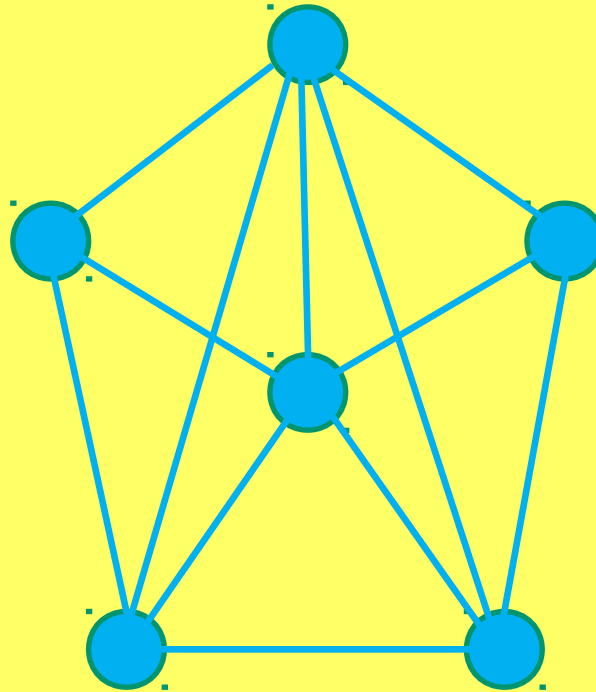


GRAF

Jest to zbiór wierzchołków, które **mogą** być połączone krawędziami, w taki sposób, że każda krawędź kończy się i zaczyna w którymś z wierzchołków. Grafy to podstawowy obiekt rozważań teorii grafów. Twórcą teoretykiem grafów był Leonarda Eulera, który rozstrzygnął zagadnienie mostów królewieckich.

Wierzchołki grafu zwykle są numerowane i czasem stanowią reprezentację jakichś obiektów, natomiast krawędzie mogą wówczas obrazować relacje między takimi obiektami. Krawędzie mogą mieć wyznaczony kierunek, a graf zawierający takie krawędzie jest **grafem skierowanym**. Krawędź może posiadać także wagę, to znaczy przypisaną liczbę. Grafy stosowane są m.in. do oznaczenia odległości na mapach pomiędzy dwoma dowolnymi miejscami lub czasu potrzebnego do przebycia tej odległości. W grafie skierowanym wagi mogą być zależne od kierunku przechodzenia przez krawędź, co stosuje się np. na mapach terenów górskich, na którym czas marszu pod górę jest inny niż z góry). Specyficznym typem grafu jest drzewo.

GRAF – prezentacja graficzna



TABLICA

Jest to struktura danych, w której poszczególne dane są dostępne dzięki tzw. kluczom (indeksom), przyjmującym najczęściej wartości numeryczne. Rozmiar tablicy jest albo ustalony z góry (tablice statyczne), albo może się zmieniać w trakcie wykonywania programu (tablice dynamiczne).

Praktycznie wszystkie języki programowania obsługują tablice. W matematyce odpowiednikiem tablicy jednowymiarowej jest ciąg, a tablicy dwuwymiarowej - macierz.

Tablica jednowymiarowa (wektor)

Służy do przechowywania w pamięci określonej (skończonej) liczby obiektów tego samego typu

Przykład:

10-cio elementowa tablica liczb całkowitych o **nazwie T**

1	2	3	4	5	6	7	8	9	10
-8	7	-20	1	7	9	-34	12	5	3

Odwołanie się do i -tego elementu tablicy T: **T[i]**, np.

```
wypisz(T[i]); //wypisz wartość i-tego elementu tablicy T  
T[i] = 8;     //i-temu elementowi tablicy T przypisz wartość 8
```

Tablice dwuwymiarowe (macierze)

Służą do przechowywania w pamięci określonej (skończonej) liczby obiektów tego samego typu

Przykład:

Tablica o nazwie **M**, o 5-ciu wierszach i 10-ciu kolumnach i elementach całkowitych

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4			27							
5										

Odwołanie się do elementu o indeksie i oraz j tablicy M : $M[i][j]$, np.

`wypisz(M[i][j]); //wypisz wartość i-tego elementu tablicy M`

`M[i][j] = 27; //elementowi tablicy M o indeksie i i j przypisz
//wartość 27`

Jeżeli $n = m$, gdzie n – liczba wierszy, m – liczba kolumn tablicy dwuwymiarowej, to **tablicę** nazywamy **kwadratową**.

STRUKTURA LUB REKORD

Jest to złożony typ danych, grupujący logicznie powiązane ze sobą dane różnego typu w jednym obszarze pamięci. Składowe struktury - **pola** - są etykietowane, tj. mają swoje unikatowe nazwy; poprzez podanie nazwy otrzymuje się dostęp do danego pola.

Struktury są powszechnie stosowane w programowaniu, pozwalając w przejrzysty sposób opisywać złożone obiekty. Przykładem struktury może być informacja o książce, której pola będą zawierały: imię i nazwisko autora (typ łańcuchowy), tytuł (typ łańcuchowy), rok wydania (liczba całkowita), liczba stron (liczba całkowita), nazwa wydawnictwa, numer ISBN itp.

KONIEC