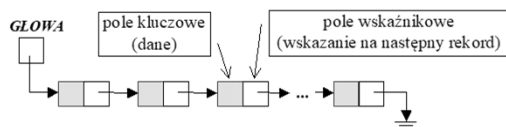


## Struktury dynamiczne

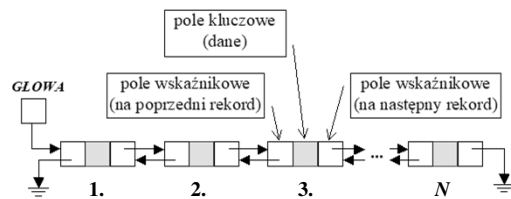
### LISTY WSKAŹNIKOWE:

- listy z dowiązaniem jednokierunkowymi,
- listy z dowiązaniem dwukierunkowymi,
- listy cykliczne z wartownikiem ...

Lista jednokierunkowa:

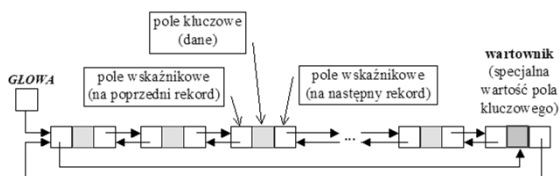


Lista dwukierunkowa:



Kolejność rekordów na liście jest jednoznacznie ustalona przez liczbę wskazań (adresów) jakie należy odczytać zaczynając od głowy listy np. w celu poznania zawartości ich pola kluczowego.

Lista dwukierunkowa cykliczna z wartownikiem:



Algorytm sumowania płac wszystkich pracowników firmy, których dane zapamiętano w liście jednokierunkowej:

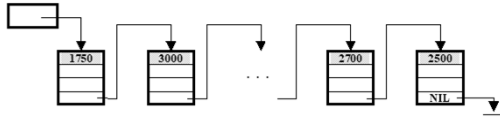
### Użyte struktury danych

Rekord do przechowania danych pojedynczego pracownika:

pole liczbowe <i>PLACA</i> -	2700
pole tekstowe <i>IMIE</i> -	Jan
pole tekstowe <i>NAZWISKO</i> -	Dynamiczny
pole wskaźnikowe <i>NASTĘPNY</i> -	→

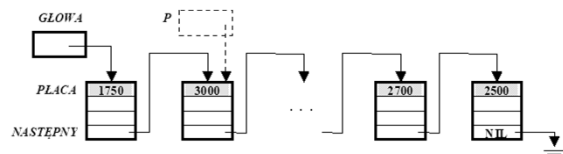
Lista do przechowania danych wszystkich pracowników:

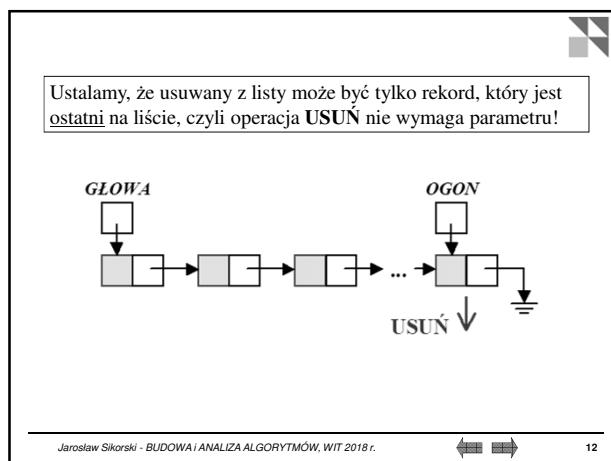
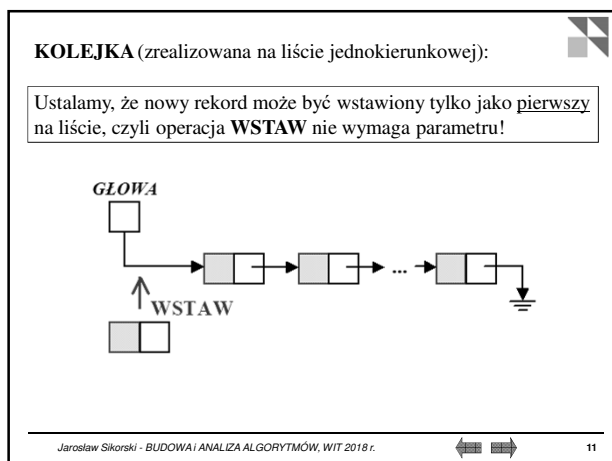
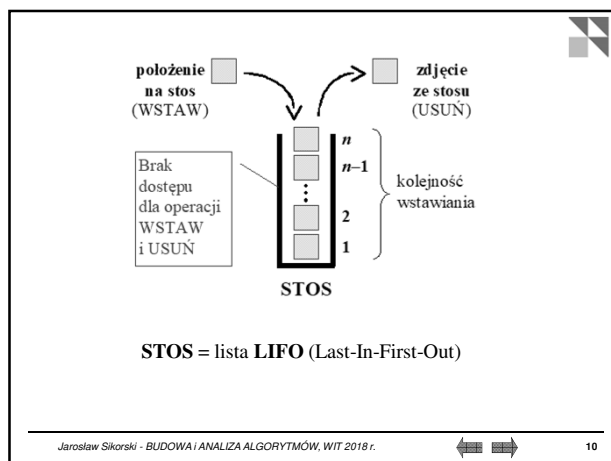
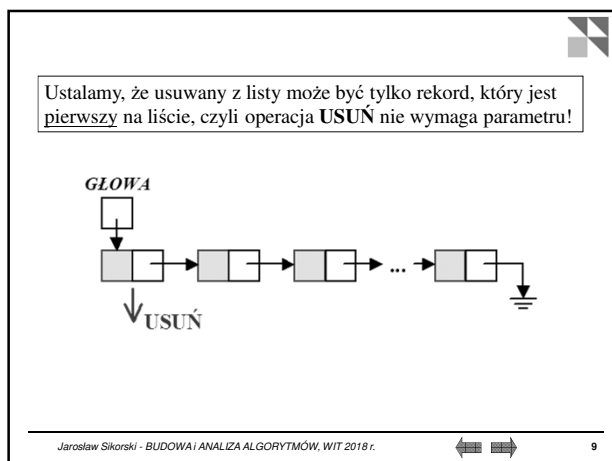
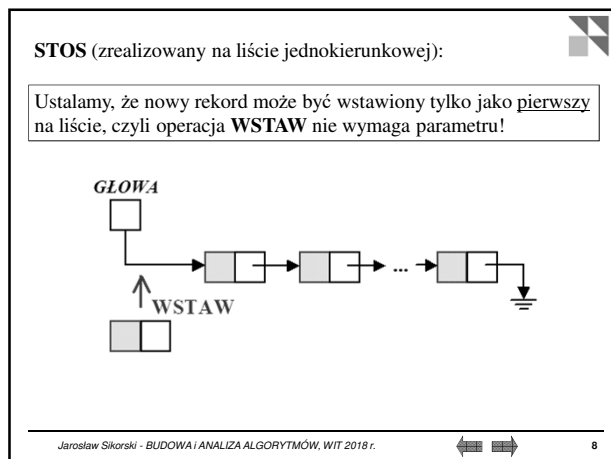
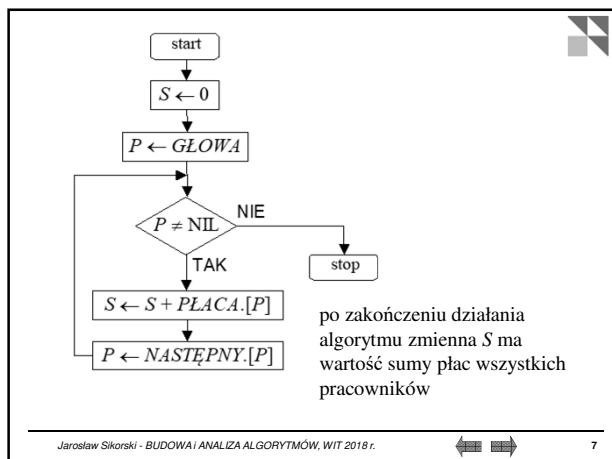
GŁOWA (wskaźnik)

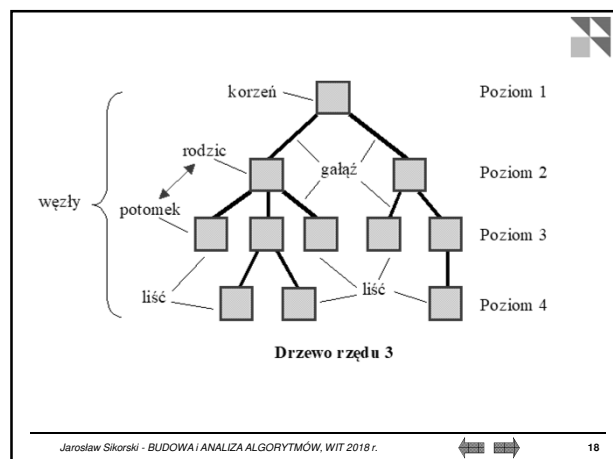
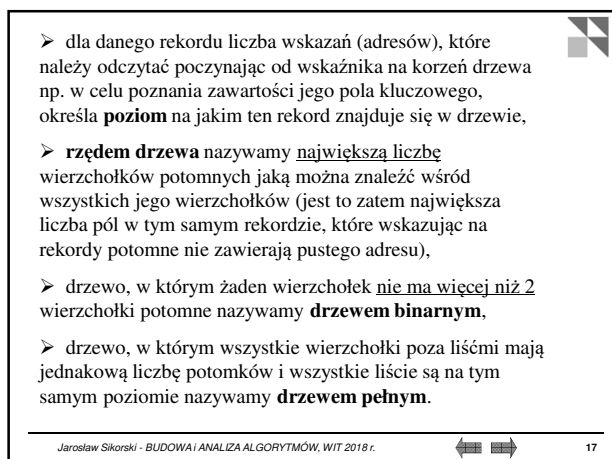
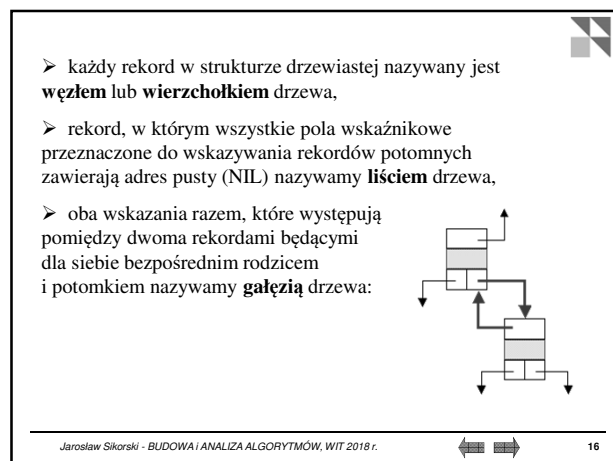
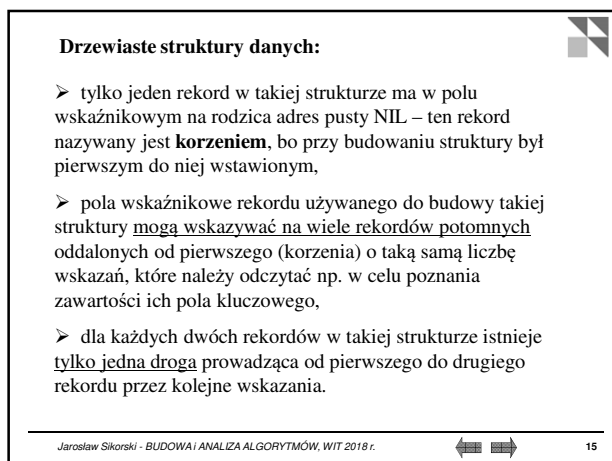
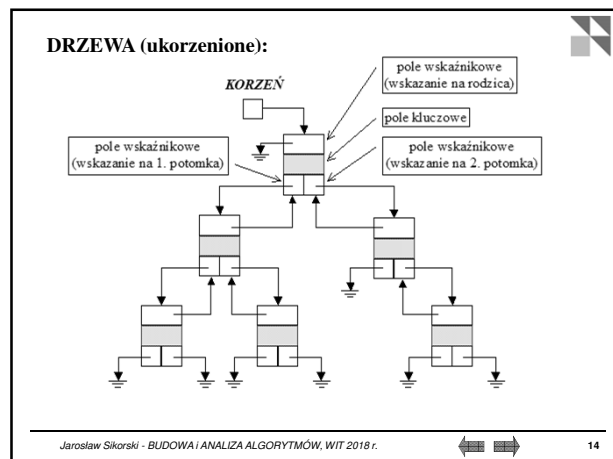
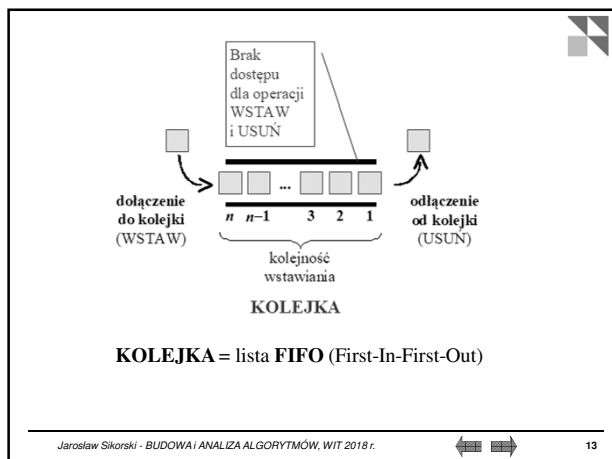


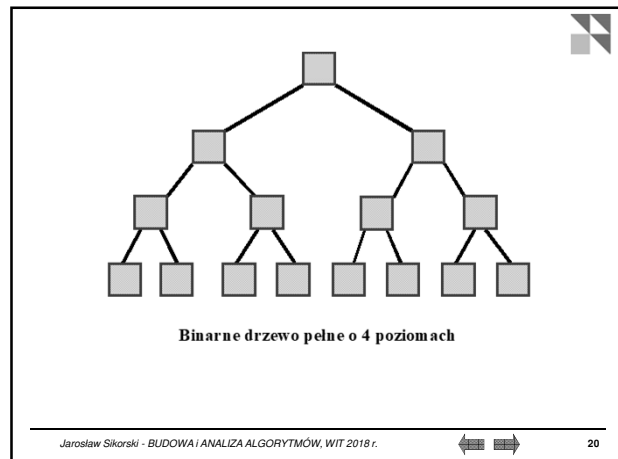
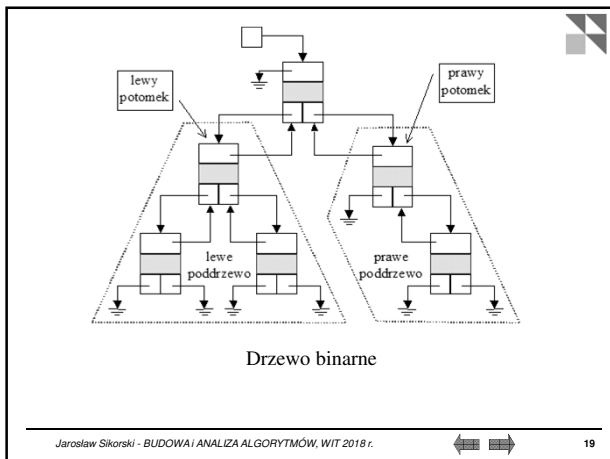
Zmienna wskaźnikowa *P* do wskazywania bieżącego rekordu z danymi pracownika,  
pomocnicza zmienna *S* do przechowywania wyniku.

- $S \leftarrow 0$  (ustalenie początkowej wartości sumy płac),
- $P \leftarrow GŁOWA$   
(ustalenie początkowej wartości bieżącego wskaźnika),
- dopóki  $P \neq NIL$  wykonuj co następuje:
  - $S \leftarrow S + PLACA.[P]$ ,
  - $P \leftarrow NASTĘPNY.[P]$ ,
- odczytaj wartość zmiennej *S*.









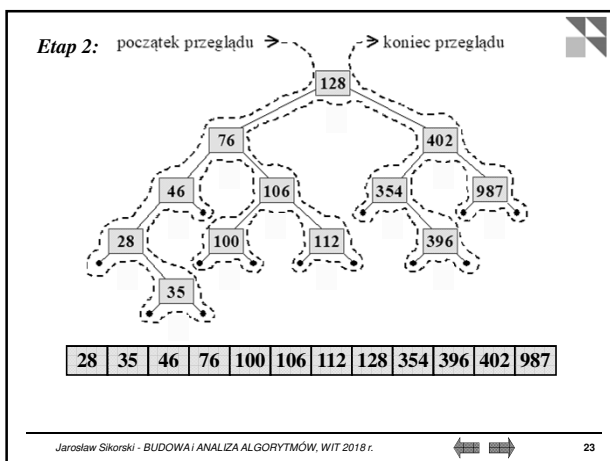
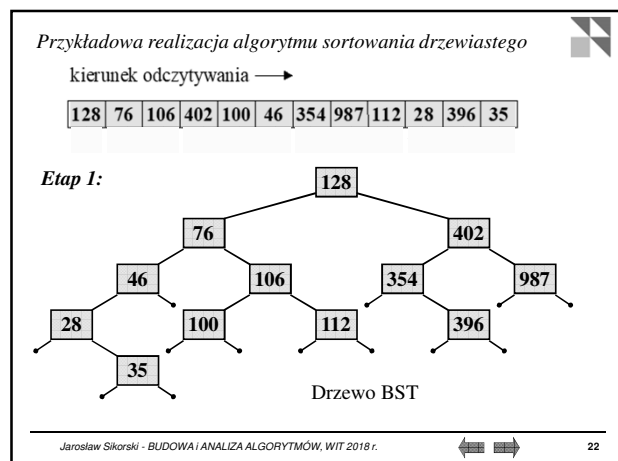
**Algorytm sortowania drzewiastego:**

**Etap 1.** Zapisanie elementów z nieuporządkowanej listy wejściowej w wierzchołkach binarnego drzewa poszukiwań  $T$  (ang. Binary Search Tree)

**Etap 2.** Obejście drzewa  $T$  (odwiedzenie wszystkich wierzchołków) według zasady lewostronnego przeglądu w głąb i wypisywanie elementów listy przy drugich odwiedzinach wierzchołka.

**Drzewo BST** to takie drzewo binarne, w którym dla dowolnie wskazanego wierzchołka spełnione są dwa warunki:  
żaden z elementów zapisanych w wierzchołkach jego lewego poddrzewa nie jest większy od elementu zapisanego w tym wierzchołku i żaden z elementów zapisanych w wierzchołkach jego prawego poddrzewa nie jest mniejszy od tego elementu.

Jarosław Sikorski - BUDOWA I ANALIZA ALGORYTMÓW, WIT 2018 r.



**Procedura rekurencyjna realizująca 2. etap algorytmu sortowania drzewiastego:**

**Procedura *obejdz*( $T$ )**

- jeśli drzewo  $T$  jest puste, to wróć do poziomu wywołania,
- w przeciwnym przypadku wykonaj co następuje:
  - wywołaj *obejdz* ( $L_T$ ),
  - wypisz element umieszczony w korzeniu drzewa  $T$ ,
  - wywołaj *obejdz* ( $P_T$ ),
- wróć do poziomu wywołania.

**Reguła wypisywania IN-ORDER**

Jarosław Sikorski - BUDOWA I ANALIZA ALGORYTMÓW, WIT 2018 r.