

Algorytmy i struktury danych

Wykład 5

Elementarne struktury danych

prof. dr hab. inż. Andrzej Obuchowicz

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

23 listopada 2016

Spis treści

- 1 Podstawowe definicje
 - Struktura danych
 - Słownik
 - Zbiór liniowo uporządkowany
- 2 Stos
- 3 Kolejka
- 4 Listy
 - Lista jednokierunkowa
 - Lista jednokierunkowa cykliczna
 - Lista dwukierunkowa
 - Lista dwukierunkowa cykliczna
- 5 Już za tydzień na wykładzie

Struktura danych

Struktura danych

$$S = (D, R, e),$$

gdzie:

- $D = \{d_i\}$ — zbiór danych elementarnych;
- $e \in D$ — identyfikator struktury;
- $R = \{r_{io} = \langle e, d_i \rangle\} \cup \{r_{in} = \langle d_j, d_i \rangle\}$
— r_{io} : relacja wejściowa; r_{in} : relacja wewnętrzna.

Słownik

Słownik: struktura danych, dla której zdefiniowano następujące algorytmy

- $x = \text{find}(e, k)$
wyszukaj adres x elementu z polem $\text{key} = k$ w strukturze wskazanej przez e ;
- $\text{insert}(e, x)$
wstaw element wskazany przez x do struktury wskazanej przez e ;
- $\text{delete}(e, x)$
usuń element wskazany przez x ze struktury wskazanej przez e .

Zbiór liniowo uporządkowany

Zbiór liniowo uporządkowany: Struktura danych, której elementy (rekordy) zawierają pole typu porządkowego (tzn. zdefiniowano dla tego typu relację mniejszości), według którego można porządkować liniowo elementy zbioru. Można zdefiniować algorytmy:

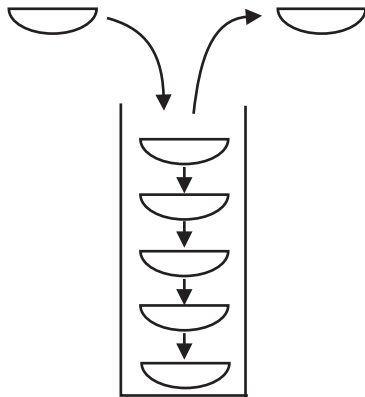
- $x = \text{next}(e, y)$ ($x = \text{prec}(e, y)$)

wyszukaj adres x elementu następującego po elemencie y (poprzedzającego element y) w strukturze wskazanej przez e ;

- $x = \text{min}(e)$ ($x = \text{max}(e)$)

wyszukaj adres x elementu minimalnego (maksymalnego) w strukturze wskazanej przez e .

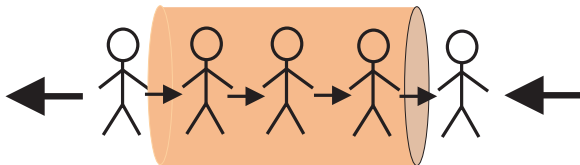
Stos (LIFO: last in first out)



```
push(top,x)
  if top<>null
    next[x] := top
  top := x

pop(top)
  if top=null
    then return 'niedomiar'
  else top := next[top]
```

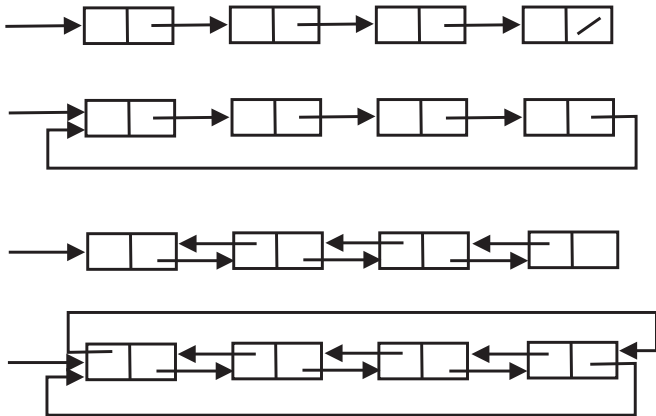
Kolejka (FIFO: first in first out)



```
enqueue(head,tail,x)
  if tail=null
    then head:=x
  else next[tail]:=x
  tail:=x
```

```
dequeue(head,tail)
  if head=null
    then return 'niedomiar'
  else if head=tail
    then head:=null
         tail:=null
  else head:=next[head]
```

Listy



Lista jednokierunkowa - operacje

```
insert_list1(head,x)
  if head<>null
    then next[x]:=head
  head:=x

x=search_list1(head,k)
  x:=head
  while x<>null & key[x]<>k
    do x:=next[x]
  return x
```

```
delete_list1(head,x)
  if head=x
    then head:=next[head]
  else y:=head
    while next[y]<>x
      do y:=next[y]
    next[y]:=next[x]
```

Lista jednokierunkowa cykliczna - operacje

```
insert_list1c(head,x)
  if head<>null
    then next[x]:=next[head]
    next[head]:=x
  else next[x]:=x
    head:=x

delete_list1c(head,x)
  if next[x]=x
    then head:=null
  else y:=head
    while next[y]<>x
      do y:=next[y]
    next[y]:=next[x]
    if head=x
      then head:=next[head]

x=search_list1c(head,k)
x:=head
if head<>null
  then while next[x]<>head & key[x]<>k
    do x:=next[x]
    if key[x]<>k
      then x:=null
return x
```

Lista dwukierunkowa - operacje

```
insert_list2(head,x)
    if head<>null
        then next[x]:=head
            prev[head]:=x
    head:=x

x=search_list2(head,k)
x:=head
while x<>null & key[x]<>k
    do x:=next[x]
return x

delete_list2(head,x)
    if next[x]<>null
        then prev[next[x]]:=prev[x]
    if prev[x]<>null
        then next[prev[x]]:=next[x]
    else head:=next[x]
```

Lista dwukierunkowa cykliczna - operacje

```
insert_list2c(head,x)
  if head<>null
    then next[x]:=next[head]
       prev[x]:=head
       prev[next[head]]:=x
       next[head]:=x
  else next[x]:=x
       prev[x]:=x
       head:=x

x=search_list2c(head,k)
  x:=head
  if head<>null
    then while next[x]<>head & key[x]<>k
          do x:=next[x]
          if key[x]<>k
            then x:=null
  return x

delete_list2c(head,x)
  if next[x]=x
    then head:=null
  else prev[next[x]]:=prev[x]
       next[prev[x]]:=next[x]
  if head=x
    then head:=next[head]
```

A w następnym tygodniu między innymi

Drzewa :

- 1 drzewa przeszukwań binarnych,
- 2 drzewa AVL,
- 3 drzewa czerwono-czarne.

Dziękuję za uwagę!!!