Podstawowe struktury danych

1) Lista
2) Stos (LIFO - Last In, First Out)
3) Kolejka (FIFO, First In, First Out)

Lista - struktura danych z sekwencyjnym dostępem do jej elementów.
mamy dostęp do wybranych elementów listy (pierwszy, ostatni, ...)
poza tym możemy przeglądać listę iteracyjnie.

Podstawowe operacje na liście

bool empty() – czy lista jest pusta

~~Elem*~~ insert() – wstawianie do listy

Elem find(Elem x) – znajdź element x
(key k) element o kluczu k

delete(Elem x) – usuń z listy.
(key k)

Jeśli typ danych trzymanych na liście ma porządek, to możemy rozróżnić listy posortowane (uporządkowane) nieposortowane.

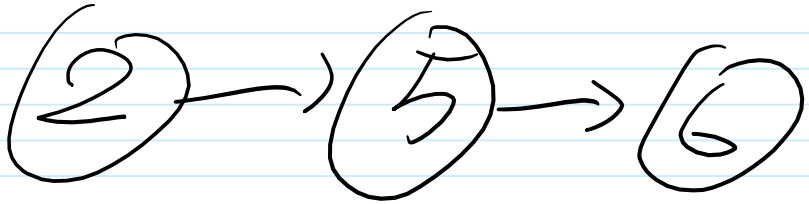
Implementacja
① w tablicy int Lista[N][2]
int head.

Lista[head][0] - to pierwszy element listy

Lista[i][1] - element kolejny po elemencie Lista[i][0].

↑ to nie musi być i-ty element listy

Jeśli Lista[i][1] = -1 to Lista[i][0] jest końcem listy



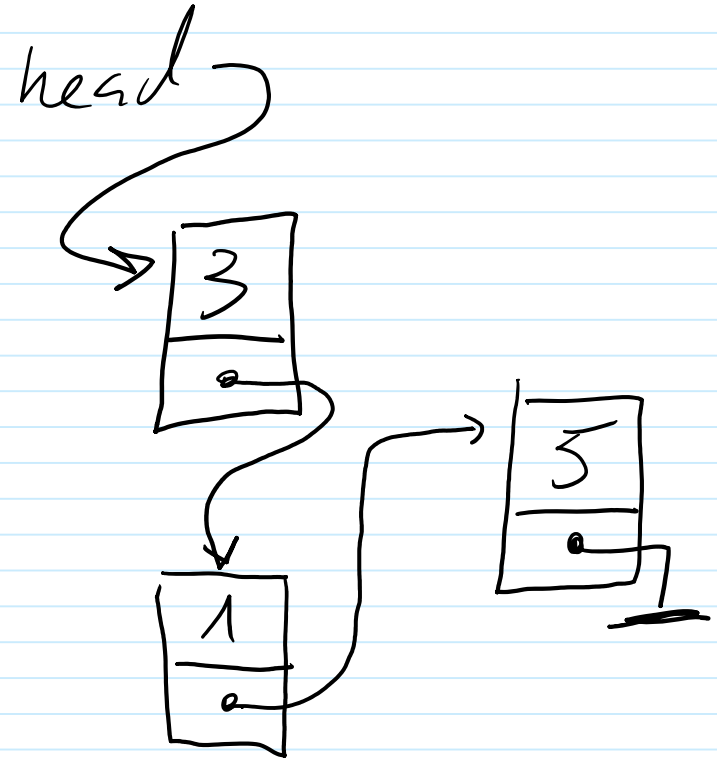
0	1	2	3	4	5	6
	6		2		5	
	-1		5		1	

head = 3

Implementacja dynamiczna

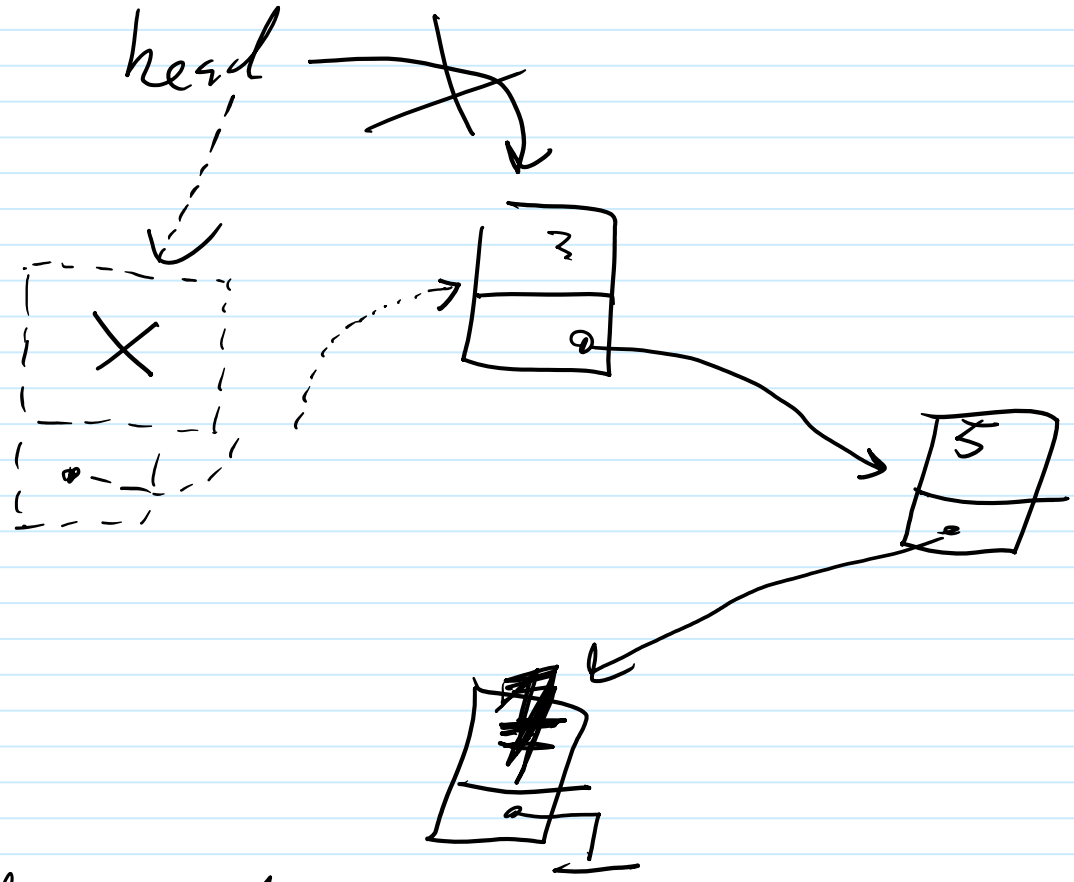
```
typedef Elem {
        int key;
        Elem * next;
} Elem;

Elem * head = NULL;
```



```
bool empty() {
    if (head == NULL) return true;
    else return false;
}
```

```
Elem * insert(int x) {
    Elem * ptr = new (Elem);
    ptr -> key = x;
    ptr -> next = head;
    head = ptr;
    return head;
}
```



Implementacja find, delete – na ćwiczeniach

Złożoność operacji na liście

	lista niepasortowana		l. pasortowana	
insert	$O(1)$	$O(1)$	$O(n)$	$\frac{n}{2}$
find	$O(n)$	$\frac{n}{2}$ – dla elementów na liście n – gdy brak szukanego elementu	$O(n)$	$\frac{n}{2}$
delete	tak samo jak find.			

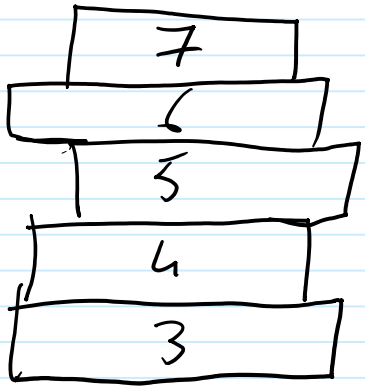
złożoność pesymistyczna

złożoność średnia

n – długość listy.

② Stos, LIFO

Last In, First Out



operacje

bool empty()

push(Elem X) - wstawienie elementu na stos

Elem pop() - zdejmuje element ze stosu i go zwraca

Elem top() - sprawdzenie co jest na wierzchołku stosu (ale bez zmiany stosu).

Logiczna specyfikacja stosu

$\forall$ s - stos $\forall$ x - element

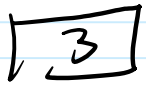
push(x) ; pop() = x

push(x) ; top() = x

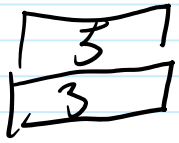
push(x) ; top() ; pop() = x

s - pusty stos

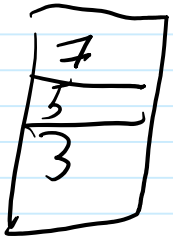
push(3)



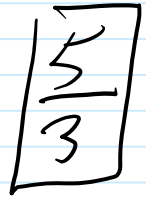
push(5)



push(7)



pop()



push(6)

Gdzie wykorzystujemy stos?

- w systemie operacyjnym - zarządzanie wywołaniami funkcji, zarządzanie rekursji
- zamiana programów rekurencyjnych na programy wykorzystujące stos
- przeglądanie struktur danych ustalanie kolejności wykonywania pewnych zadań.

Uwaga

Stos i lista udostępniają ograniczoną ilość operacji o ustalonym działaniu.

zato mamy ich stały koszt

Kolejka FIFO - First In First Out

zastosowanie

np. kolejki zadań w systemie operacyjnym

kolejki zadań do urządzeń

algorytmy (przeglądanie danych w ustalonym porządku)

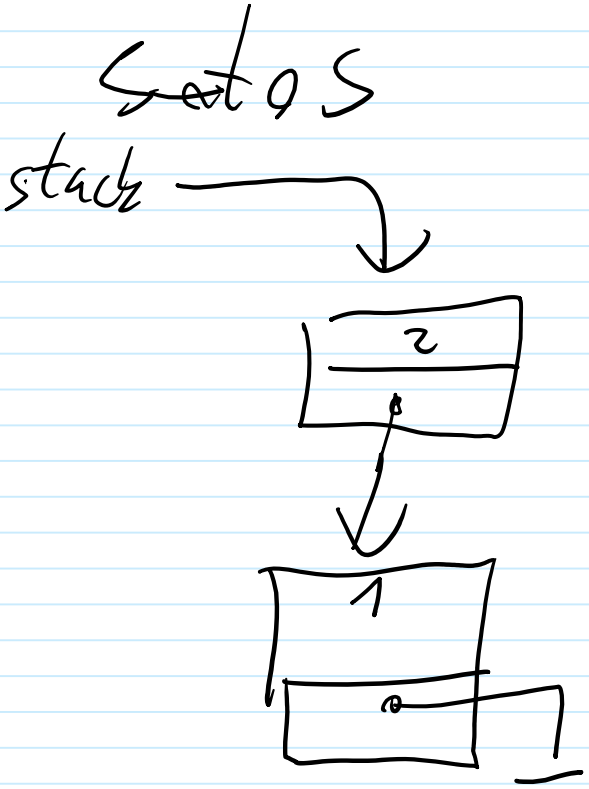
empty()

enqueue(Elem x) - dodanie na koniec kolejki

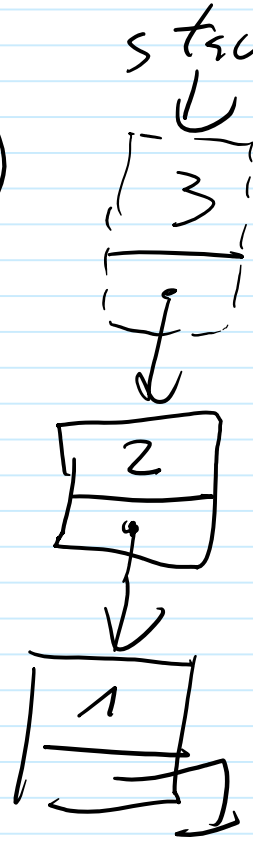
Elem dequeue() - usunięcie elementu z początku kolejki

Elem first() - sprawdzenie, co jest na początku kolejki.

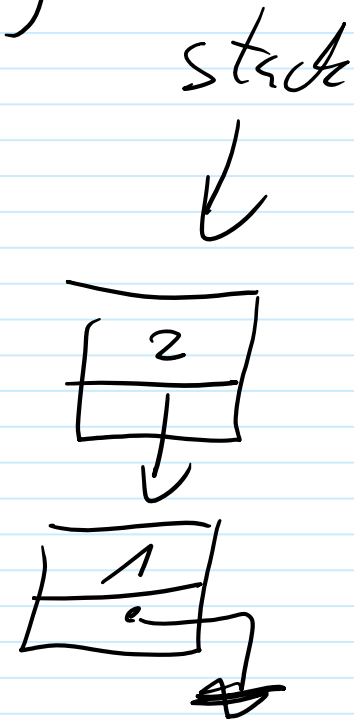
Implementacja stosu i kolejki na liście.

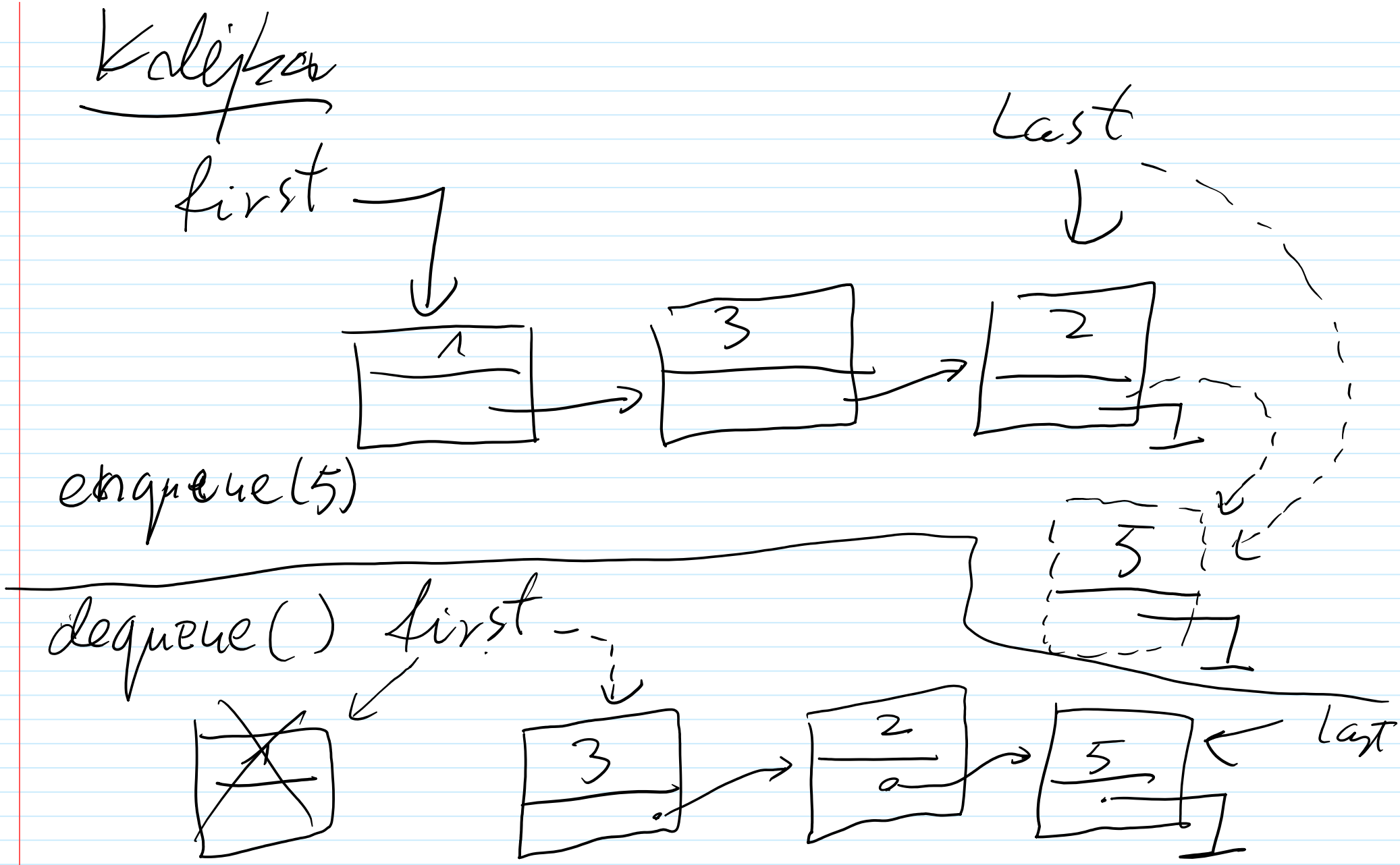


push(3)



pop()




Kolejka
first
Last
1
3
2
enqueue(5)
5
dequeue()
first
3
2
5
Last