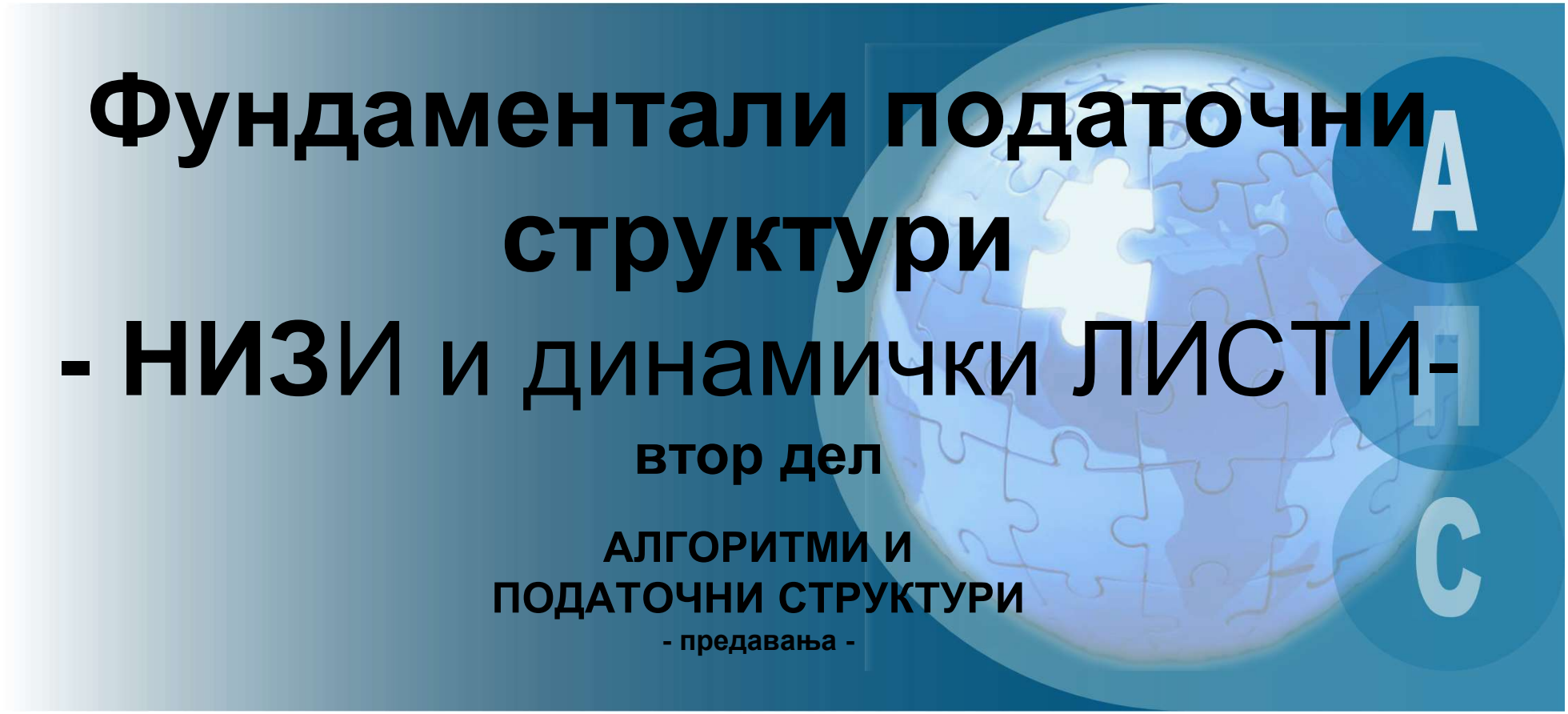
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Фундаментали податочни структури
- НИЗИ и динамички ЛИСТИ-

втор дел

**АЛГОРИТМИ И
ПОДАТОЧНИ СТРУКТУРИ**

- предавања -

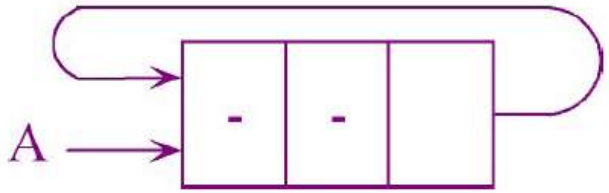


Кружни поврзани листи

- Поврзаните листи би можело да дадат комплексни алгоритми кога станува збор за нивно изминување кое содржи повеќе итерации
- Проблемот на комплексни алгоритми за SLLco почеток и крај се решава со користење на кружни поврзани листи
- Репрезентација: Нема потреба од нулев покажувач за крајот на листата. Се воведува посебен јазел - **водач** кој е почеток и крај на листата

Кружни поврзани листи

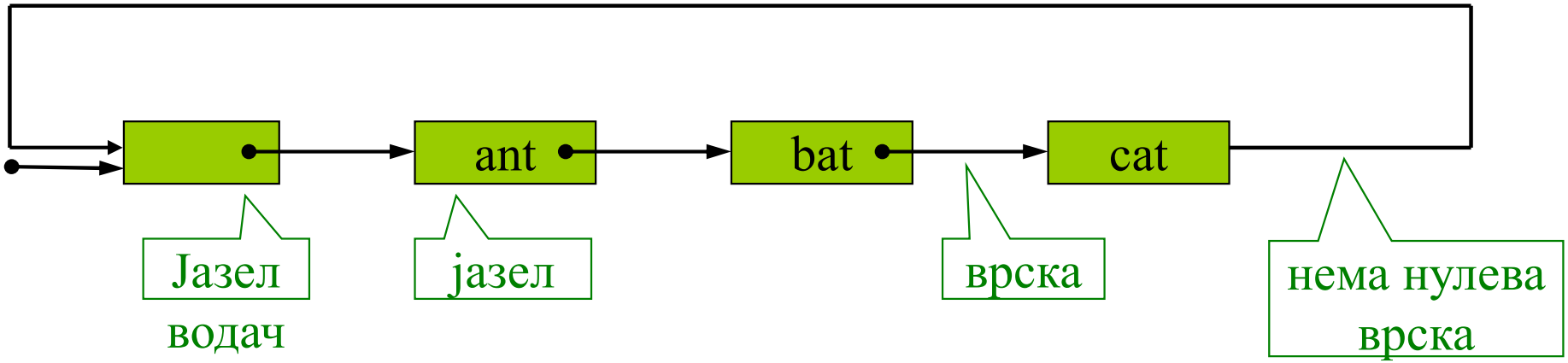
- Празна кружна листа не е одредена со нулев покажувач, туку со јазел чие инфо поле е празно, а линкот покажува на сам себе (на истиот јазел)



- Кружните листи може да се избришат во фиксно време

Кружни поврзани листи

- Изминувањето на кружна листа се прави се додека не го посетиме јазелот водач, од каде сме почнале



- Најширока употреба при распределување процеси во ОС

Кружни поврзани листи

- Во општ случај останатите операции како вметнување елемент и бришење се слични како кај едностранo поврзаните листи

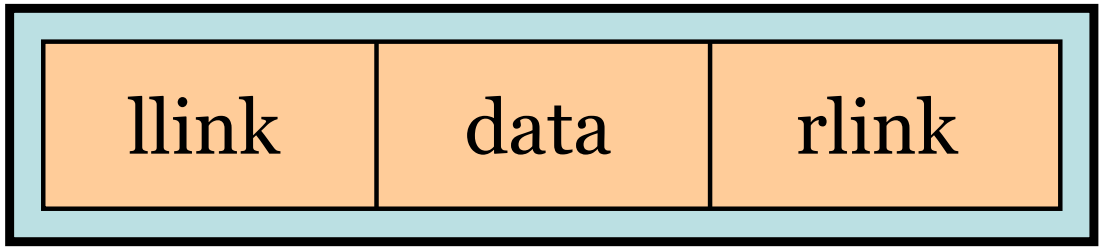
- Добар пример за самостојна работа дома

Двојно поврзани листи

- **Проблем кај поврзаните листи: Пронаоѓање на претходникот на јазел при операциите на вметнување и бришење на јазел!**
- Еднонасочност во движењето низ структурата
- **Решение на проблемот: Воведување на двонасочна можност за движење низ структурата!**

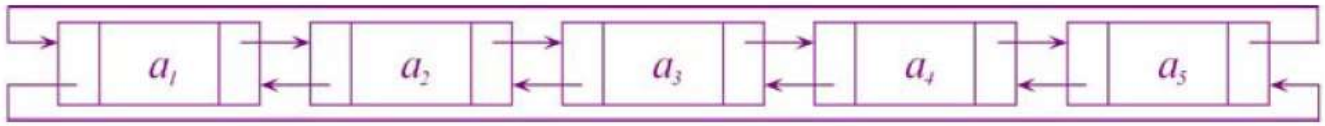
Двојно поврзани листи

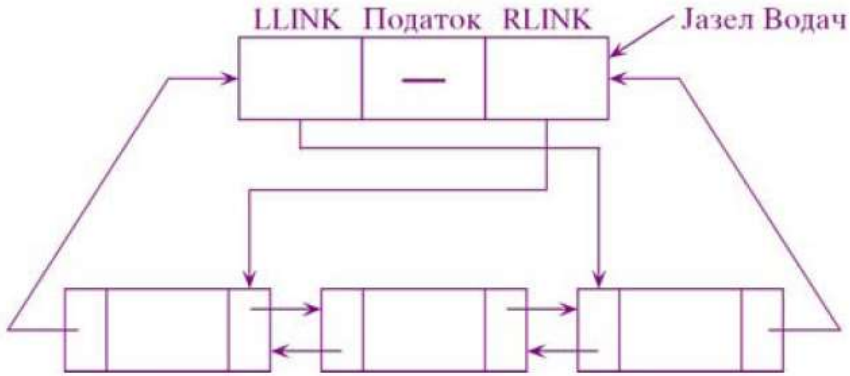
- Репрезентација: Јазелот ќе биде составен од *data* поле и **два покажувачи** - *rlink* покажувач кон следниот јазел и *llink* покажувач кон претходниот јазел во низата

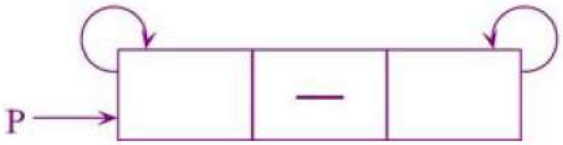


Двојно поврзани листи

- Пример репрезентации на двојно поврзани листи:





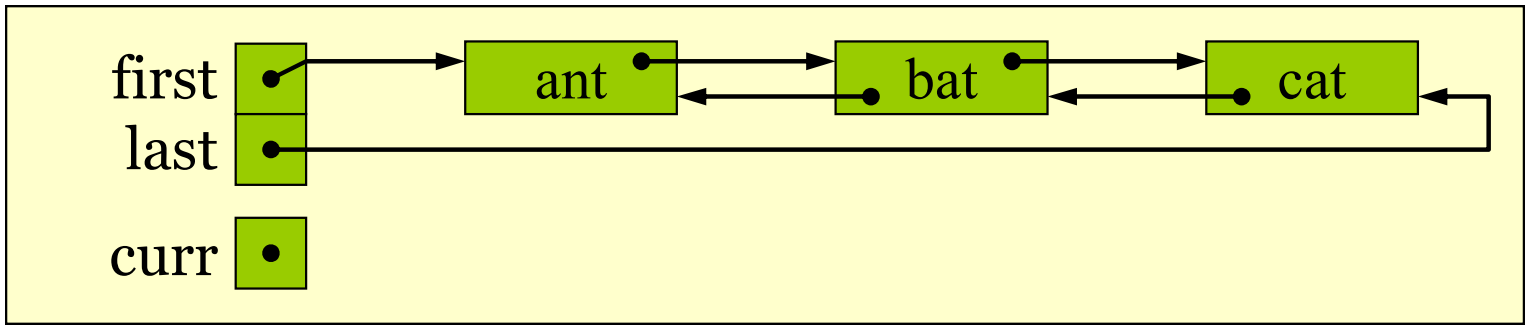


Двојно поврзани листи

- Изминување на двојно поврзана листа

```
public void printLastToFirst () {
// Print all elements in this DLL, in last-to-first order.
   for (DLLNode curr = this.last;
         curr != null; curr = curr.pred)
      System.out.println(curr.element);
}
```

- Анимација:



Користење на двојно поврзани листи

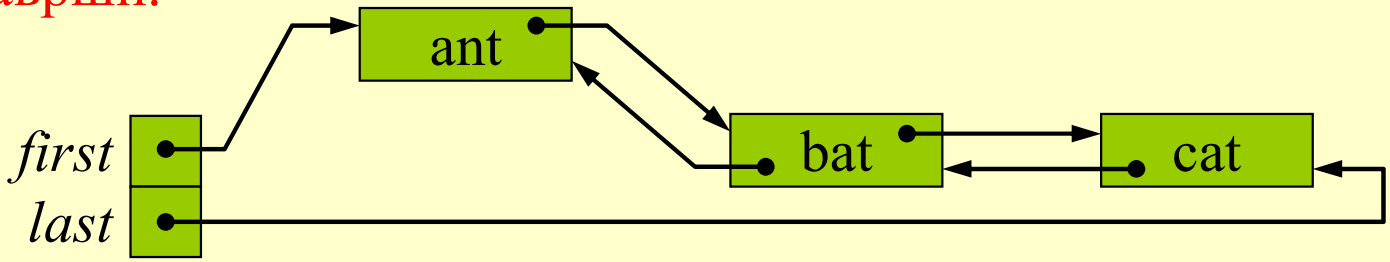
- Алгоритам за вметнување јазол во DLL:
  1. се избира јазел кој моментално не се користи во достапниот мемориски простор
  2. се доделува соодветната вредност на инфо полето од јазелот
  3. вредноста на rlink полето се пополнува со адресата на јазелот кој треба да биде следбеник на новиот јазел, а вредноста на llink полето со претходникот на новиот јазел
  4. вредноста на rlink полето на јазелот кој ќе биде претходник на новиот јазел треба да се промени со адресата на новиот јазел и вредноста на llink полето на јазелот кој ќе биде следбеник на новиот јазел да се промени со адресата на новиот јазел

Користење на двојно поврзани листи

- Анимација (вметнување пред првиот јазел):

Да се вметне *elem* во дадена DLL на почеток:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави следбеник на јазелот *ins* да биде *first* и *first* да биде *ins*
3. Нека *succ* е следбеникот на јазелот *ins*
4. Постави го претходникот на јазелот *succ* да биде *ins*.
5. Заврши.



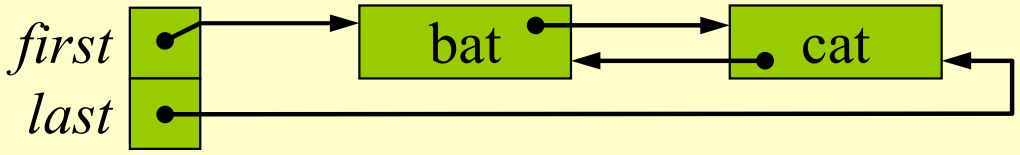
Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*

3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврши.

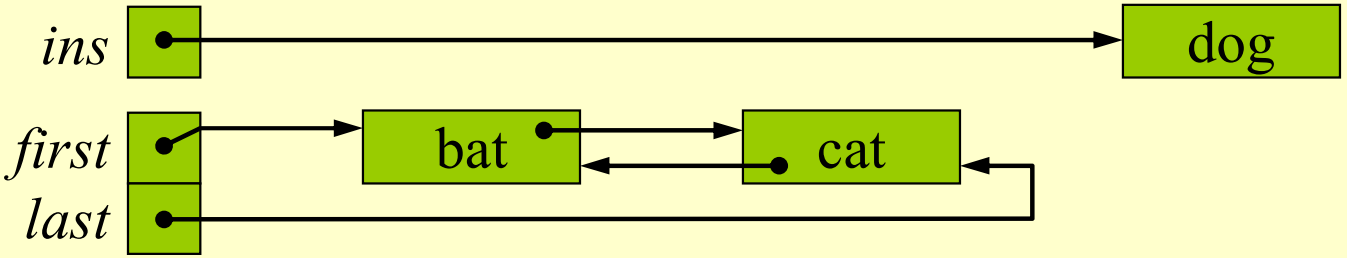


Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*
3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврши.

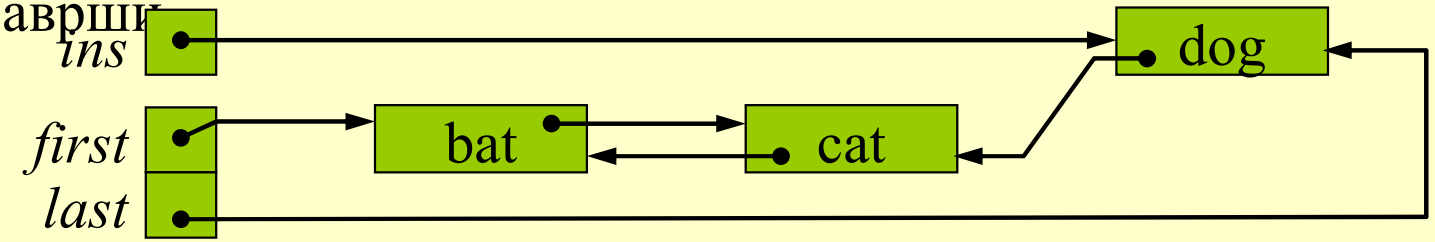


Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*
3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврш

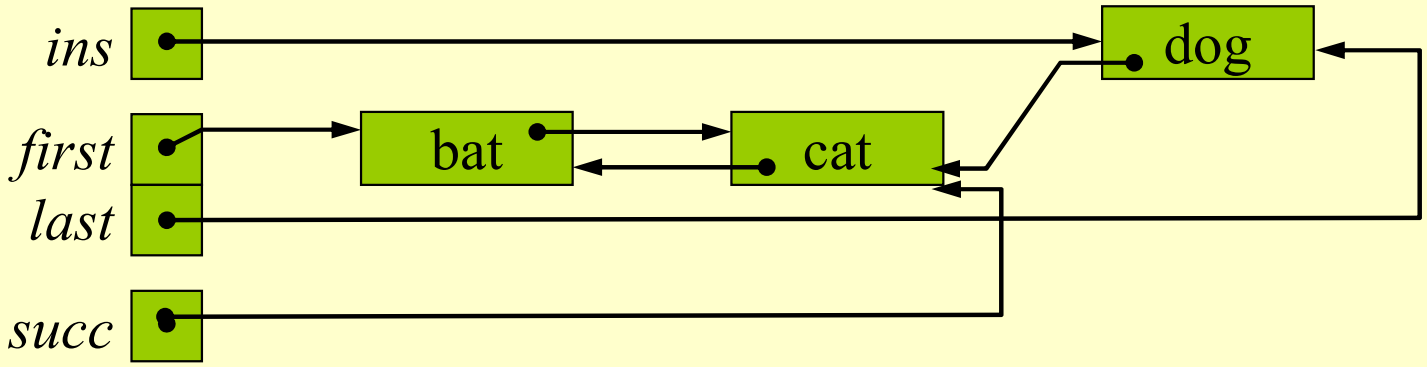


Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*
3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврши.



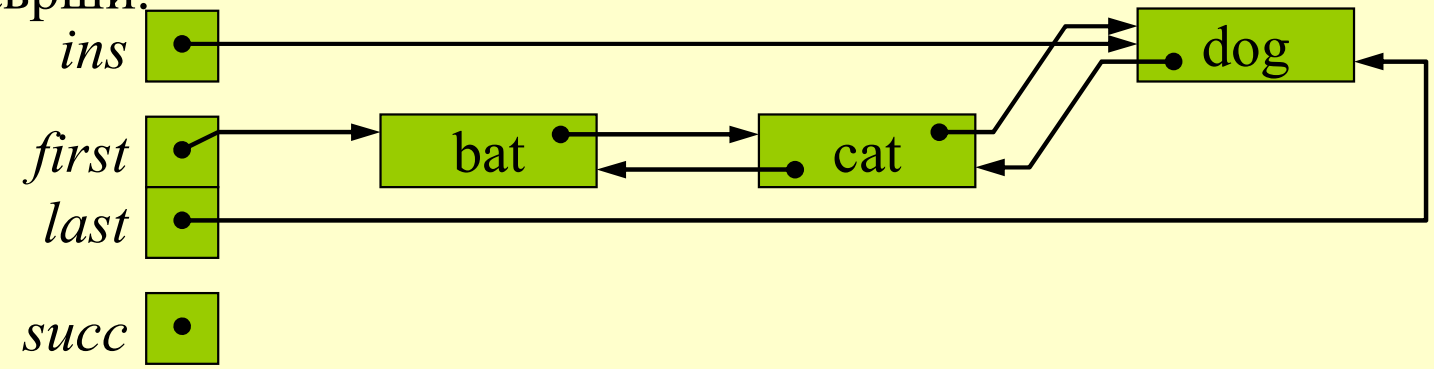
Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*

3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврши.



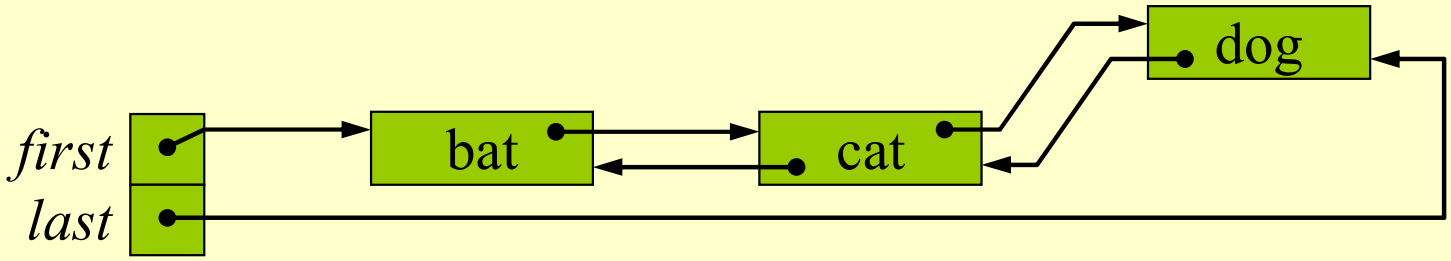
Користење на двојно поврзани листи

- Анимација (вметнување после крајниот јазел):

Да се вметне *elem* во дадена DLL на крај:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *last* и *last* да биде *ins*

3. Нека *succ* е претходникот на *ins*
4. Постави следбеник на јазелот *succ* да биде *ins*
5. Заврши.

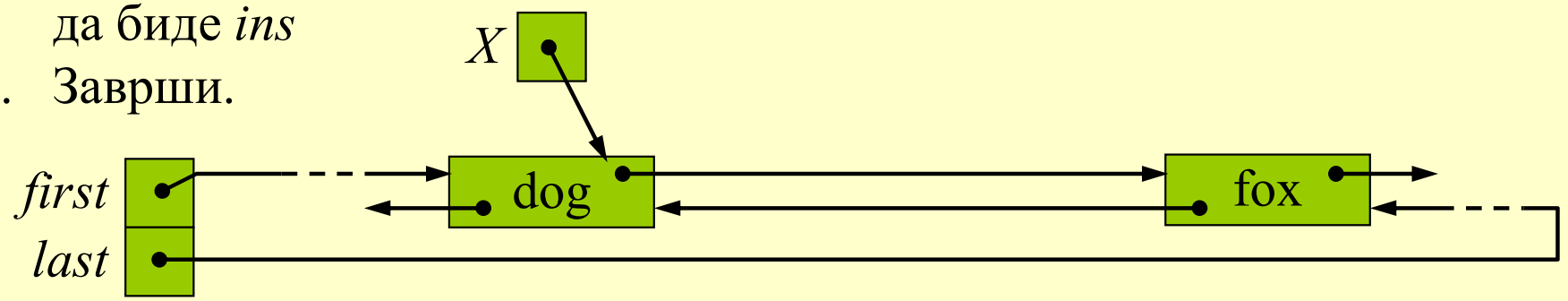


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins*, а следбеник на *X* да биде *ins*
5. Заврши.

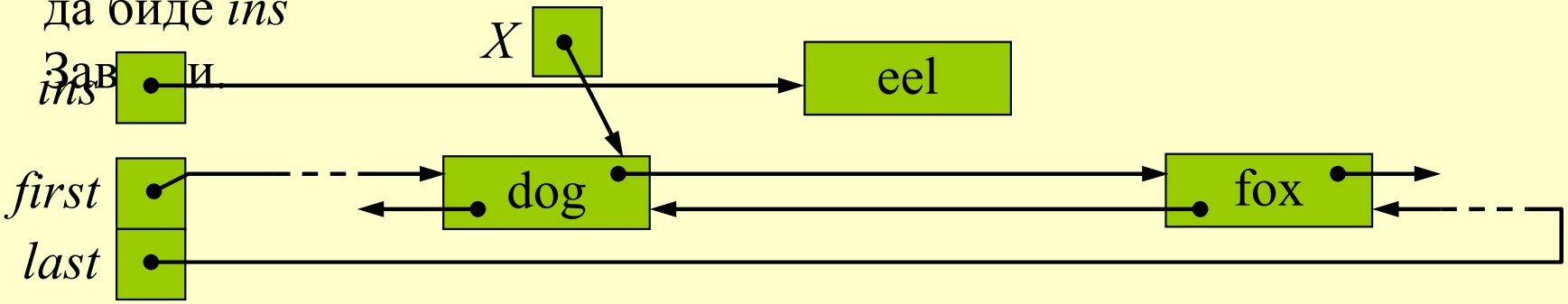


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins,* а следбеник на *X* да биде *ins*
5. Зав и.

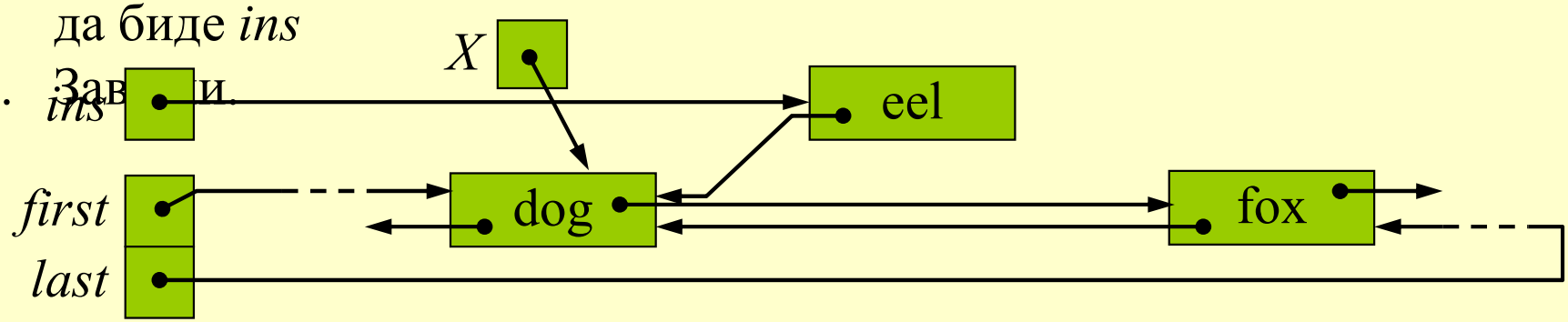


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins,* а следбеник на *X* да биде *ins*
5. Зав и.

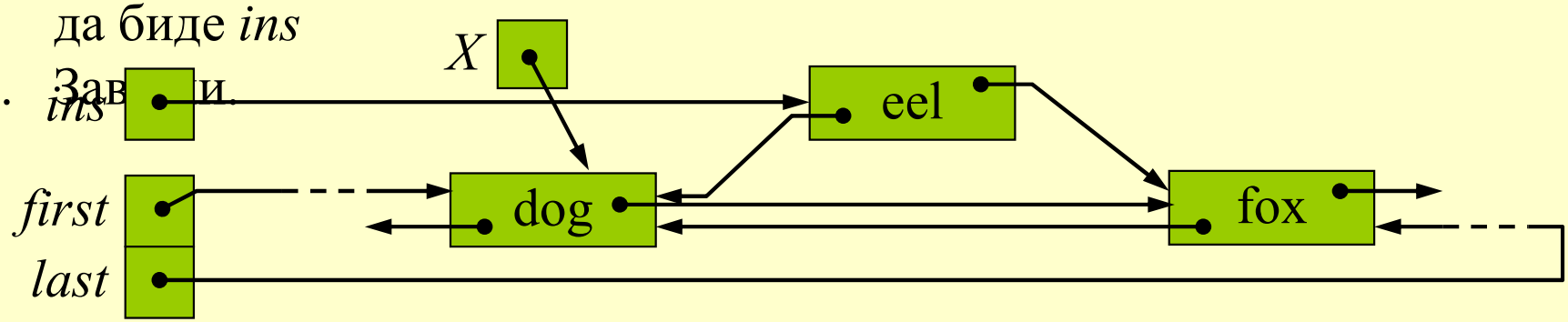


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins,* а следбеник на *X* да биде *ins*
5. Зав[illegible]и.

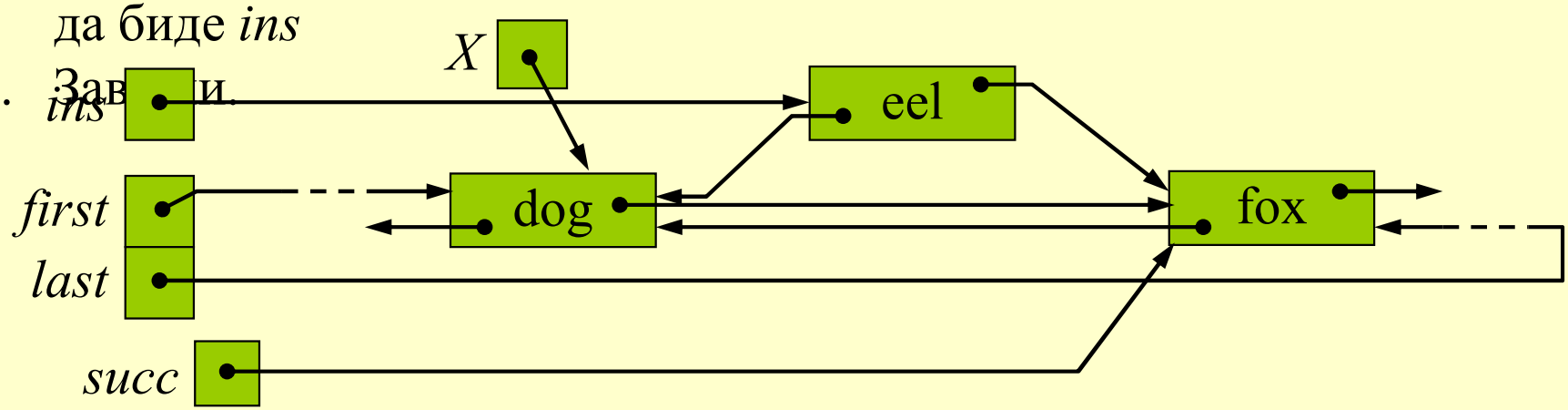


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins,* а следбеник на *X* да биде *ins*
5. Зав[illegible]и.

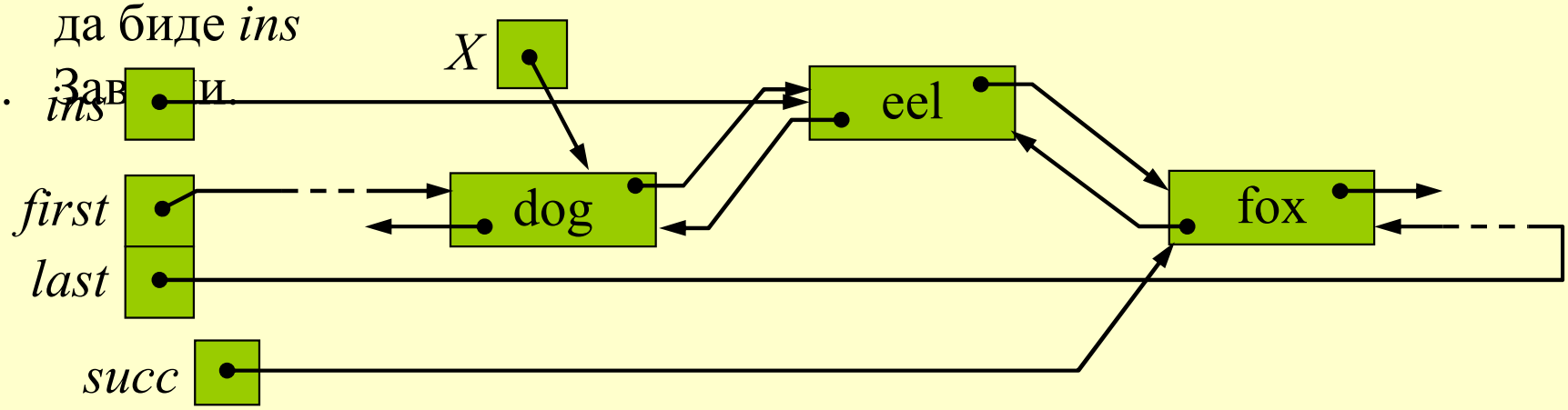


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins*, а следбеник на *X* да биде *ins*
5. Зав[illegible]и.

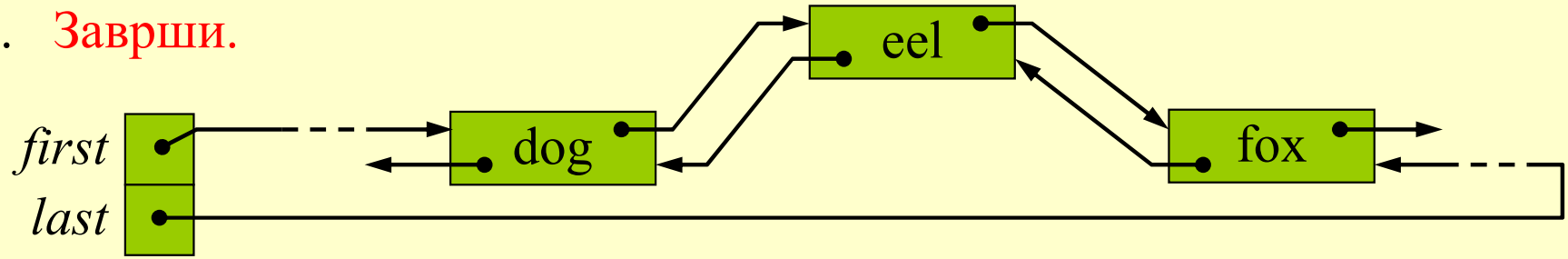


Користење на двојно поврзани листи

- Анимација (вметнување измеѓу јазли):

Да се вметне *elem* во дадена DLL после јазол X:

1. Направи го јазелот *ins* нов јазел со инфо *elem* и претходник и следбеник null.
2. Постави претходник на јазелот *ins* да биде *X*
3. Постави следбеник на *ins* да биде следбеникот на јазелот *X*
3. Нека *succ* е следбеникот на *ins*
4. Постави претходник на јазелот *succ* да биде *ins,* а следбеник на *X* да биде *ins*
5. Заврши.



Користење на двојно поврзани листи

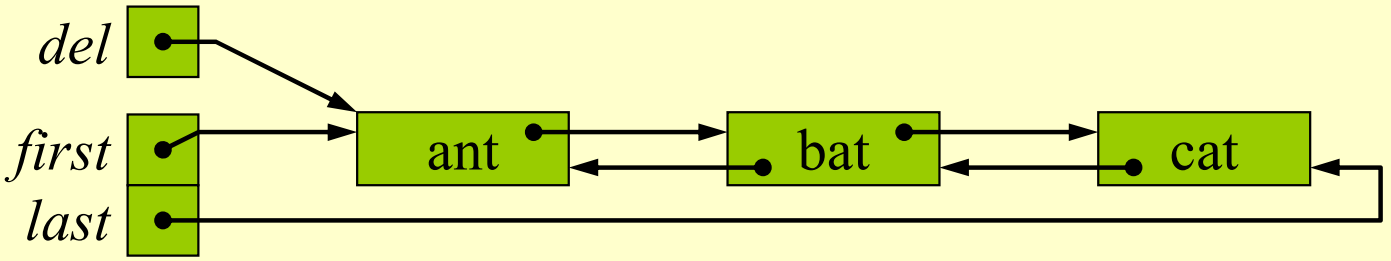
- Алгоритам за бришење јазол во DLL:
  1. се избира елемент кој сакаме да го избришеме
  2. се наоѓаат соодветно неговиот претходник и следбеник
  3. вредноста на rlink полето на неговиот претходник се променува со адресата на јазелот кој е негов следбеник
  4. вредноста на llink полето на неговиот следбеник се променува со адресата на новиот претходник

Користење на двојно поврзани листи

- Анимација (бришење на прв (но не последен) јазел)

Да се избрише јазел *del* од дадена DLL на почеток:

1. Нека *succ* е следбеник на јазелот *del*.
2. Го поставуваме *first* да покажува на јазелот *succ*.
3. Поставуваме претходник на *succ* да биде *null*
4. Заврши.

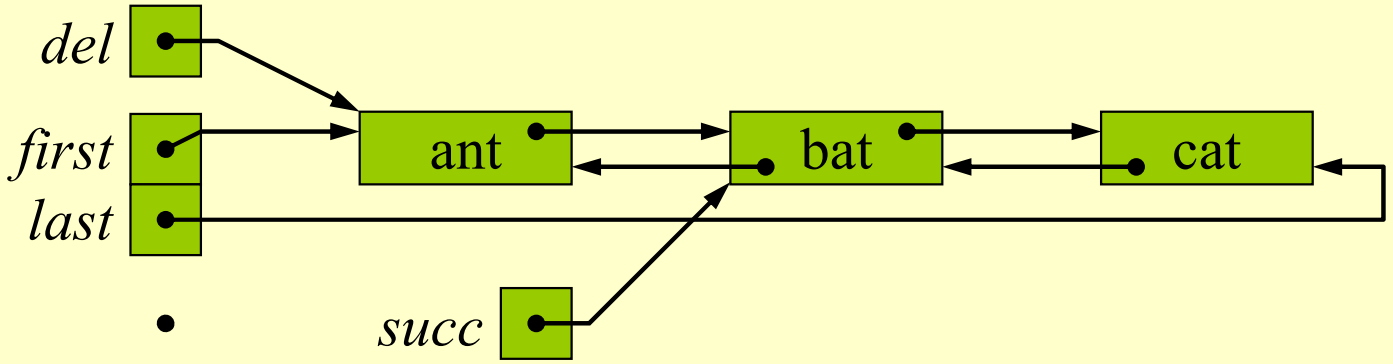


Користење на двојно поврзани листи

- Анимација (бришење на прв (но не последен) јазел)

Да се избрише јазел *del* од дадена DLL на почеток:

1. Нека *succ* е следбеник на јазелот *del*.
2. Го поставуваме *first* да покажува на јазелот *succ*.
3. Поставуваме претходник на *succ* да биде *null*
4. Заврши.

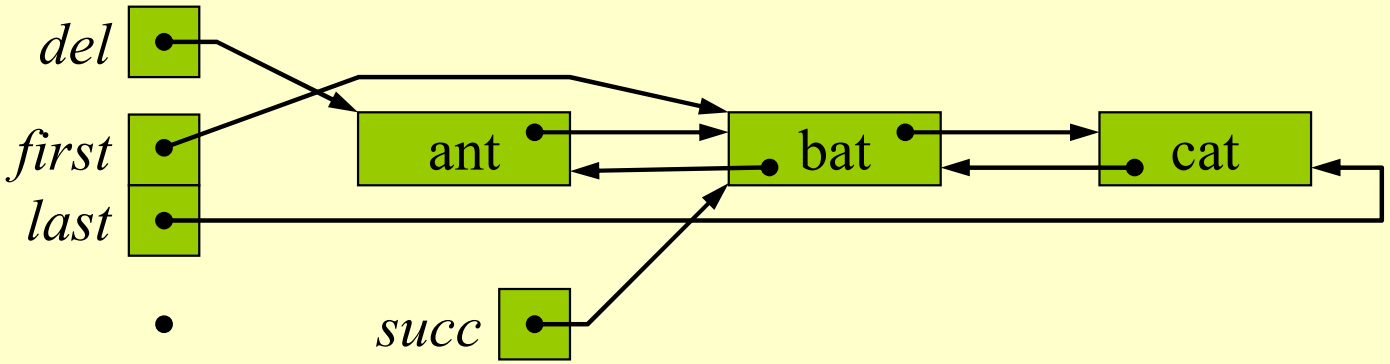


Користење на двојно поврзани листи

- Анимација (бришење на прв (но не последен) јазел)

Да се избрише јазел *del* од дадена DLL на почеток:

1. Нека *succ* е следбеник на јазелот *del*.
2. Го поставуваме *first* да покажува на јазелот *succ*.
3. Поставуваме претходник на *succ* да биде *null*
4. Заврши.

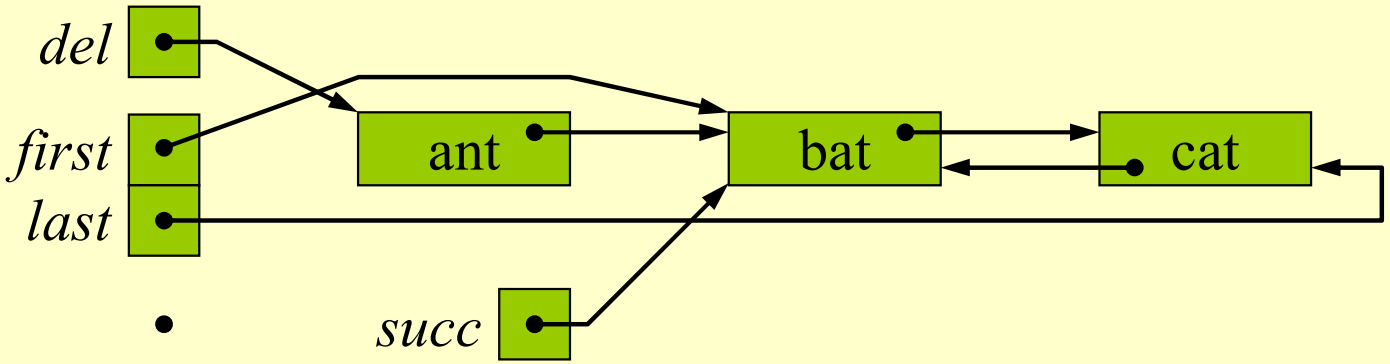


Користење на двојно поврзани листи

- Анимација (бришење на прв (но не последен) јазел)

Да се избрише јазел *del* од дадена DLL на почеток:

1. Нека *succ* е следбеник на јазелот *del*.
2. Го поставуваме *first* да покажува на јазелот *succ*.
3. Поставуваме претходник на *succ* да биде *null*
4. Заврши.

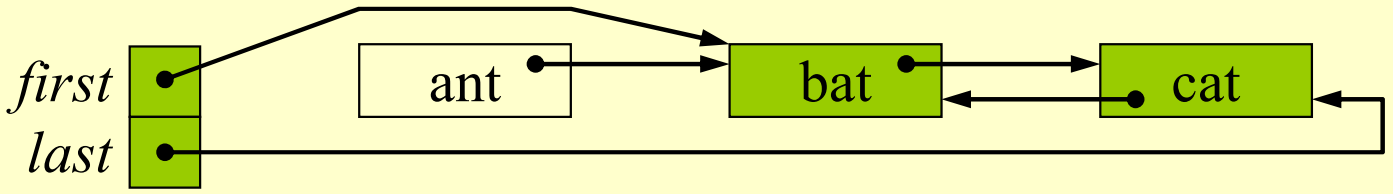


Користење на двојно поврзани листи

- Анимација (бришење на прв (но не последен) јазел)

Да се избрише јазел *del* од дадена DLL на почеток:

1. Нека *succ* е следбеник на јазелот *del*.
2. Го поставуваме *first* да покажува на јазелот *succ*.
3. Поставуваме претходник на *succ* да биде *null*
4. Заврши.

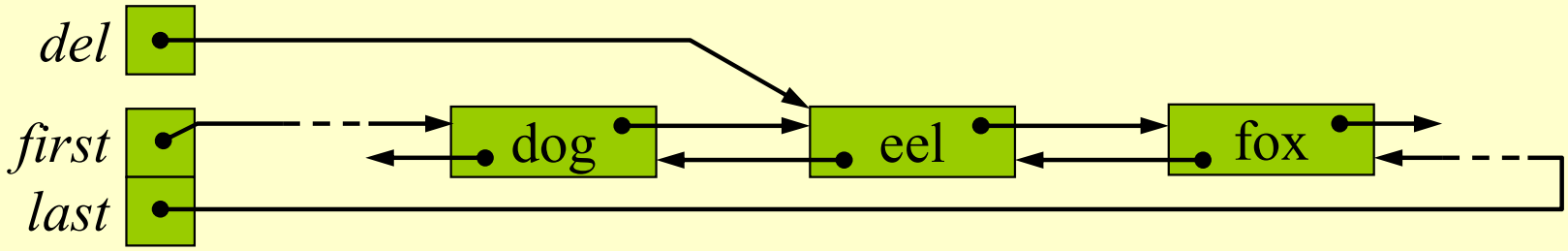


Користење на двојно поврзани листи

- Анимација (бришење на средишен јазел):

Да се избрише произволен јазел *del* од DLL:

1. Нека *pred* и *succ* се јазли кои означуваат претходник на јазелот *del* и следбеник на јазелот *del* соодветно.
2. Постави го *succ* да биде следбеник на јазелот *pred*.
3. Постави го *pred* да биде претходник на јазелот *succ*.
4. Заврши.

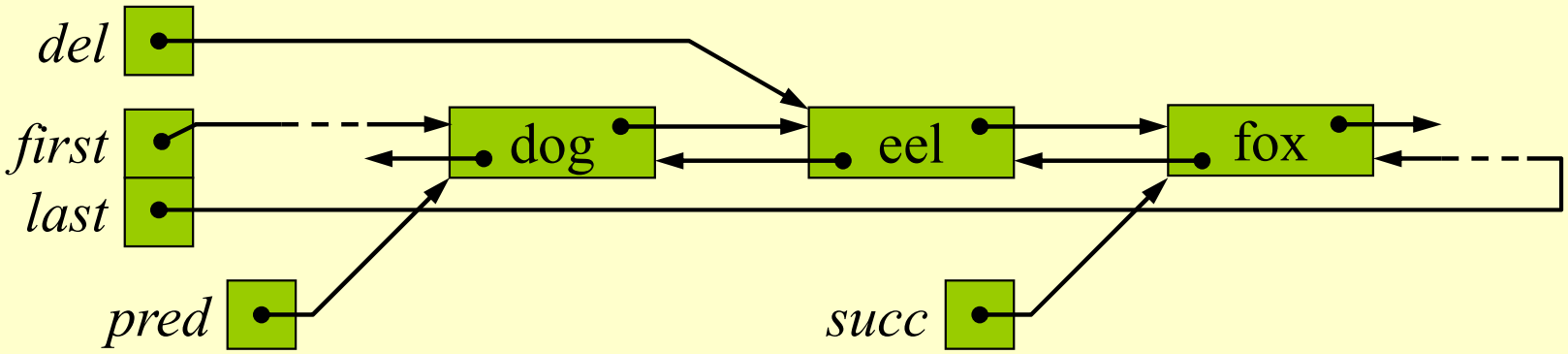


Користење на двојно поврзани листи

- Анимација (бришење на средишен јазел):

Да се избрише произволен јазел *del* од DLL:

1. Нека *pred* и *succ* се јазли кои означуваат претходник на јазелот *del* и следбеник на јазелот *del* соодветно.
2. Постави го *succ* да биде следбеник на јазелот *pred*.
3. Постави го *pred* да биде претходник на јазелот *succ*.
4. Заврши.

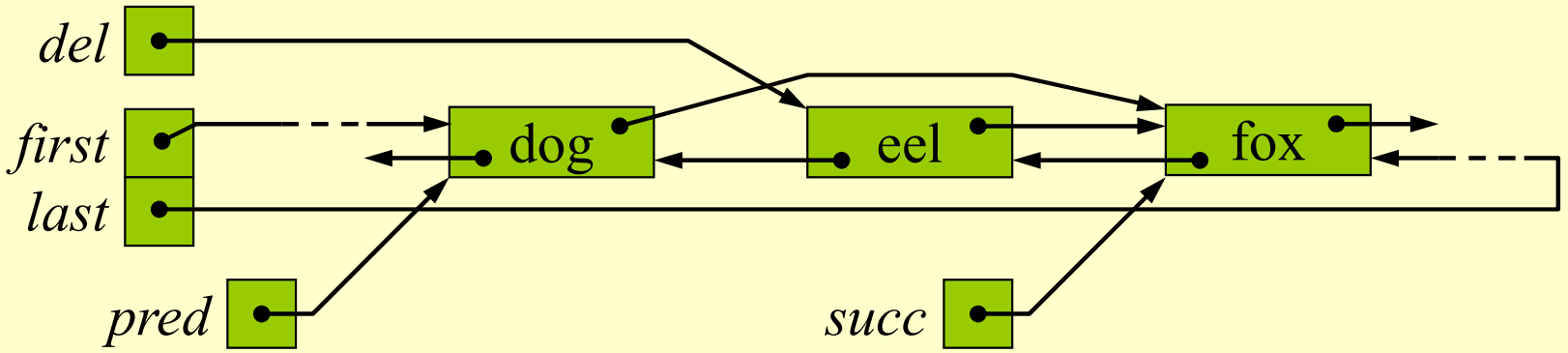


Користење на двојно поврзани листи

- Анимација (бришење на средишен јазел):

Да се избрише произволен јазел *del* од DLL:

1. Нека *pred* и *succ* се јазли кои означуваат претходник на јазелот *del* и следбеник на јазелот *del* соодветно.
2. Постави го *succ* да биде следбеник на јазелот *pred*.
3. Постави го *pred* да биде претходник на јазелот *succ*.
4. Заврши.

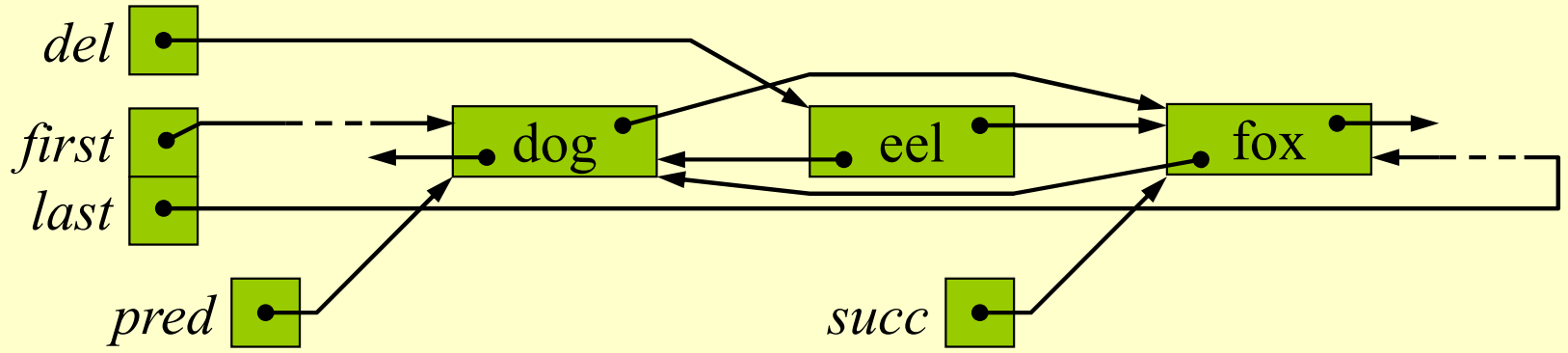


Користење на двојно поврзани листи

- Анимација (бришење на средишен јазел):

Да се избрише произволен јазел *del* од DLL:

1. Нека *pred* и *succ* се јазли кои означуваат претходник на јазелот *del* и следбеник на јазелот *del* соодветно.
2. Постави го *succ* да биде следбеник на јазелот *pred*.
3. Постави го *pred* да биде претходник на јазелот *succ*.
4. Заврши.

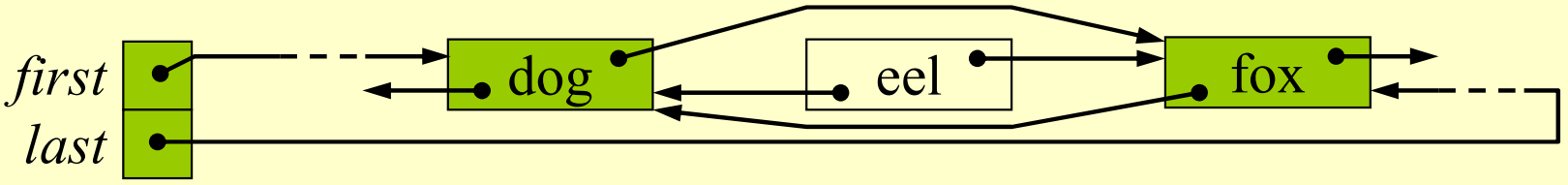


Користење на двојно поврзани листи

- Анимација (бришење на средишен јазел):

Да се избрише произволен јазел *del* од DLL:

1. Нека *pred* и *succ* се јазли кои означуваат претходник на јазелот *del* и следбеник на јазелот *del* соодветно.
2. Постави го *succ* да биде следбеник на јазелот *pred*.
3. Постави го *pred* да биде претходник на јазелот *succ*.
4. Заврши.



Најчести грешки при работа со динамичка алокација на меморијата!

- Погрешно користење на покажувачите кон мемориските локации!

- Единствени коректни операции со покажувачите кон јазлите кај листите може да бидат:
  - проверка дали се нулеви
  - проверка дали се еднакви со вредноста на други покажувачи (дали покажуваат на ист јазел)

Сложеност на операциите

Операција	Несортирана SLL	Несортирана DLL
Пребарување	$O(n)$	$O(n)$
Додавање	$O(1)$	$O(1)$
Бришење	$O(n)$	$O(1)$
Претходник	$O(n)$	$O(1)$
Следбеник	$O(1)$	$O(1)$
Минимум	$O(n)$	$O(n)$
Максимум	$O(n)$	$O(n)$