

IB111

Programování a algoritmizace

Datové struktury I

Zřetězené seznamy

Opakování – struktura (záznam)

- ```
struct student {
 int vek;
 float studijni_prumer;
 char jmeno[30];
} st, *uk_st;
```
- ```
st.vek=22; st.studijni_prumer=1.12;  
strcpy (st.jmeno, "Jan Kos");
```
- ```
uk_st = &st; uk_st ->vek=23;
```

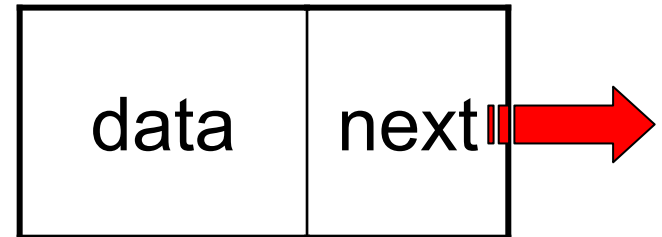


# Pole

- Výhody
  - Jednoduchý přístup přes indexy
- Nevýhody
  - Náročné přesouvání prvků
  - Fixní počet prvků
  - Přidávání a mazání prvků
    - Např. v setříděné podobě

# Zřetěžený seznam

- Datová struktura spolu s ukazatelem na další prvek
  - Ukazatel ukazuje na příští nebo předchozí prvek
- ```
struct item {  
    int data;  
    struct item *next; };
```

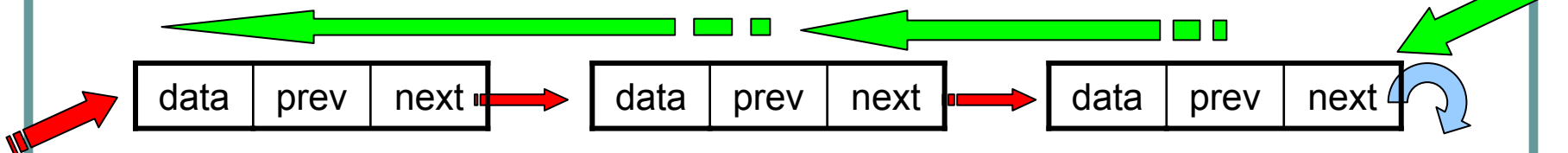


Typy zřetězených seznamů

- Jednosměrně zřetězený seznam



- Obousměrně zřetězený seznam



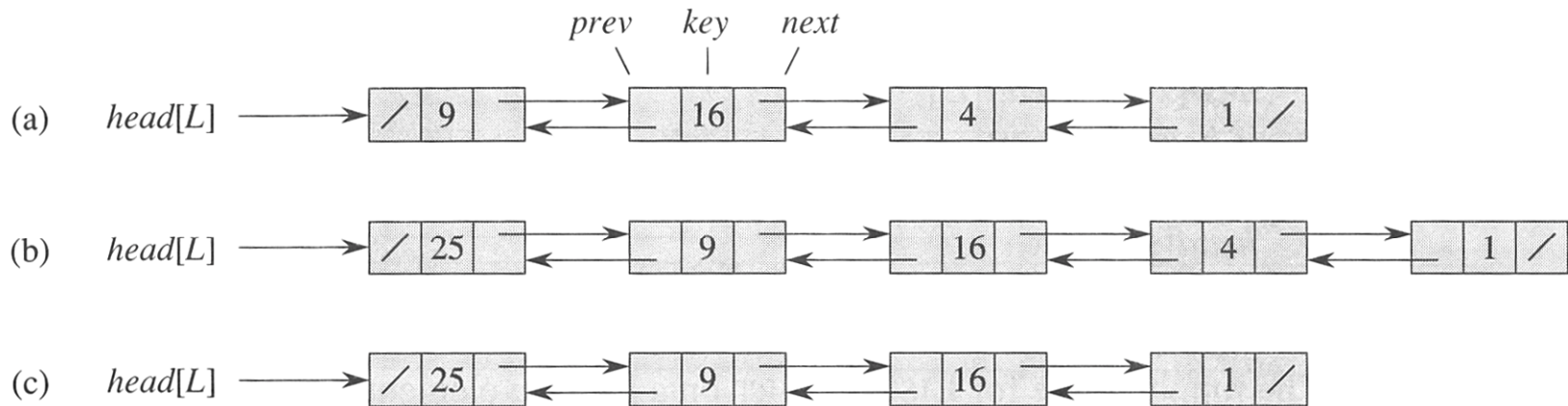
- Kruhový seznam

- Jednosměrný nebo obousměrný



Obousměrně zřetězený seznam

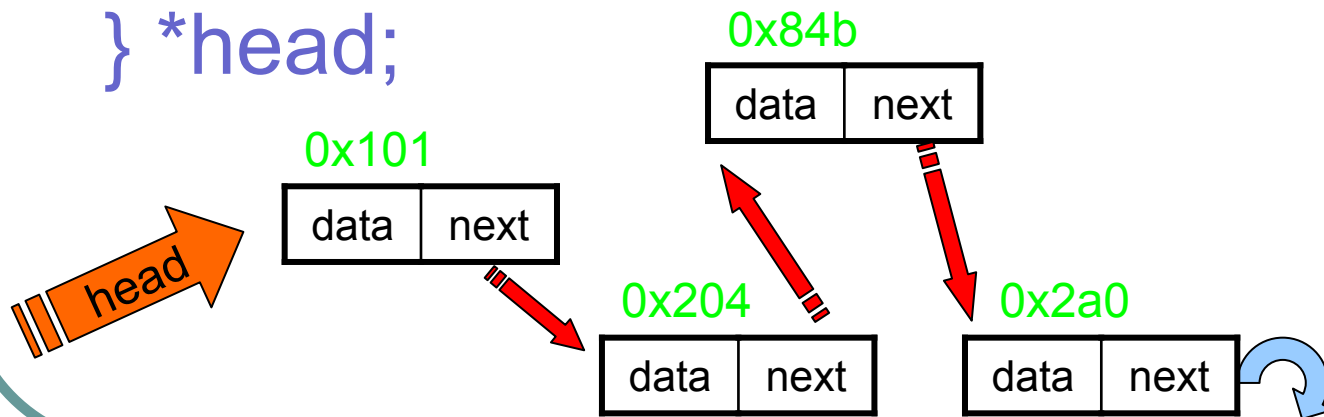
● Příklad



Praktický příklad v C

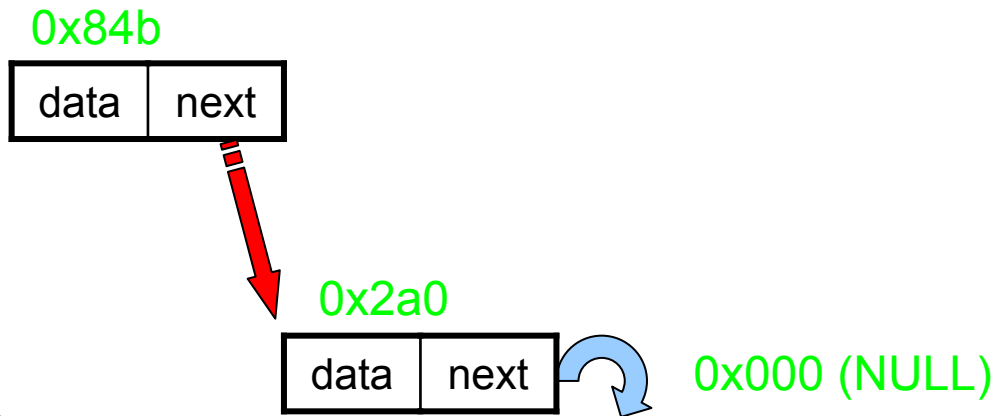
```
● struct student {  
  int vek;  
  float studijni_prumer;  
  char jmeno[30];  
  struct student *next;  
} *head;
```

Data nejsou uložena v paměti kontinuálně, ale na různých adresách podle toho jak jsme prvky přidávali/alokovali.



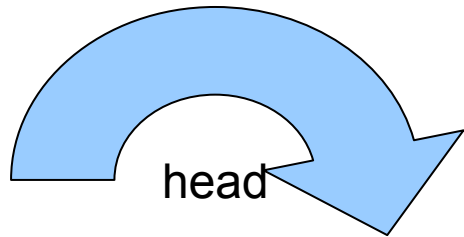
Konec seznamu

- Jak zakončit zřetěžený seznam
 - Ukazatel na další prvek nastavit na NULL
 - Přidat speciální prvek, který nebude obsahovat žádnou hodnotu
 - Kruhový seznam



Př. Prázdný seznam

- `struct student *head;`
`head = NULL;`



0x000 (NULL)

Př: Seznam s jedním prvkem

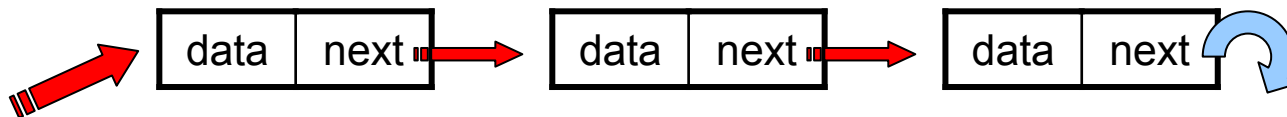
- `struct student *n = (struct student*) malloc(sizeof(struct student));`
`if (n==NULL) exit(1);`
- `n->vek = 20; n->studijni_prumer=1.0;`
`strcpy(n->jmeno, "Jan Kos");`
`n->next=NULL;`
- `struct student *head = n;`

Př: přidáme druhý prvek

- `n = (struct student*) malloc(sizeof(struct student));`
`if (n==NULL) exit(1);`
- `n->vek = 21; n->studijni_prumer=1.1;`
`strcpy(n->jmeno, "Jarek Klaus");`
`n->next=NULL;`
- `head->next = n;`

Procházení seznamu

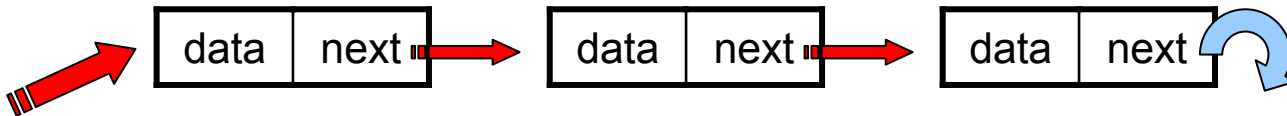
- Úkol: projít (např. vypsát) všechny prvky seznamu
 - Začneme u prvku, na který ukazuje *head*
 - Opakujeme dokud další prvek není NULL
 - Zpracuj prvek (např. vypiš hodnotu data)
 - běž na další prvek



Hledání v neseříděném seznamu

- Podobné jako procházení, ale porovnávám hodnotu s klíčem

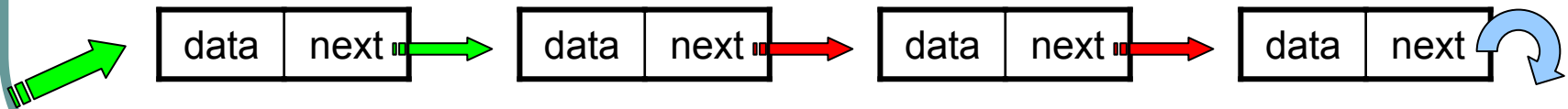
```
LIST-SEARCH(L, k)  
1  x ← head[L]  
2  while x ≠ NIL and key[x] ≠ k  
3      do x ← next[x]  
4  return x
```



Přidávání prvku

- Alokujeme paměť pro další prvek, naplníme jeho datovou část a zařadíme do seznamu
 - Na začátek
 - Na konec
 - Časová náročnost

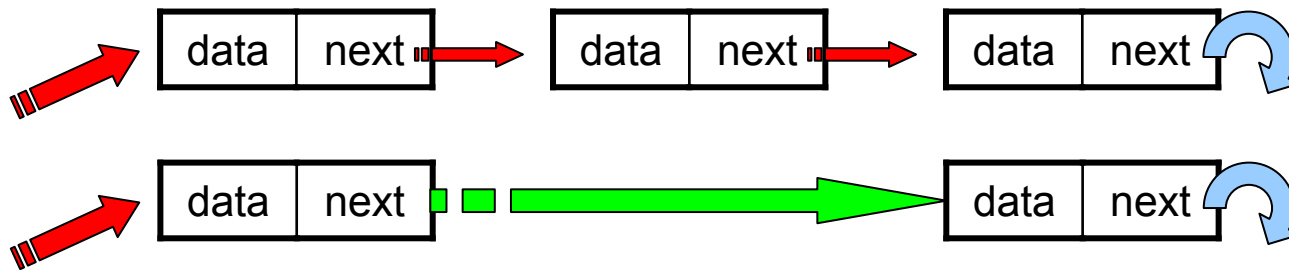
```
LIST-INSERT(L, x)
1  next[x] ← head[L]
2  if head[L] ≠ NIL
3     then prev[head[L]] ← x
4  head[L] ← x
5  prev[x] ← NIL
```



Odebrání prvku

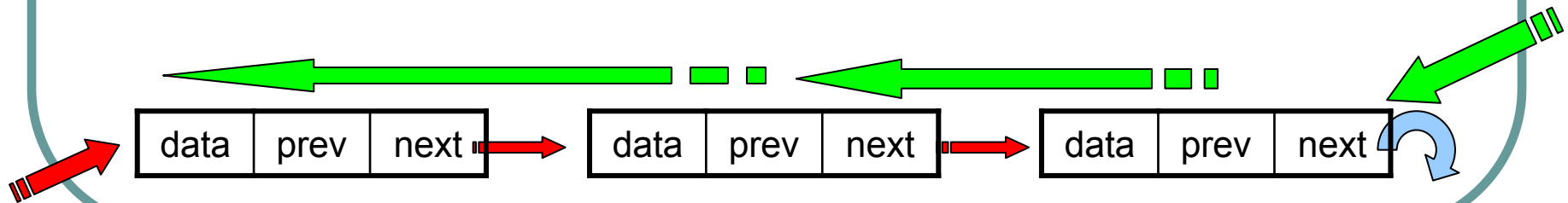
- Musíme odpojit prvek ze seznamu, dále můžeme uvolnit paměť použitou pro prvek

```
LIST-DELETE(L, x)
1  if prev[x] ≠ NIL
2      then next[prev[x]] ← next[x]
3      else head[L] ← next[x]
4  if next[x] ≠ NIL
5      then prev[next[x]] ← prev[x]
```



Oboustranně zřetězené seznamy

- Máme ukazatel na začátek a konec seznamu
 - Lehce můžeme přistoupit k prvnímu i poslednímu prvku
 - Ale musíme vždy aktualizovat oba směry
 - Přidávání prvků
 - Mazání prvků

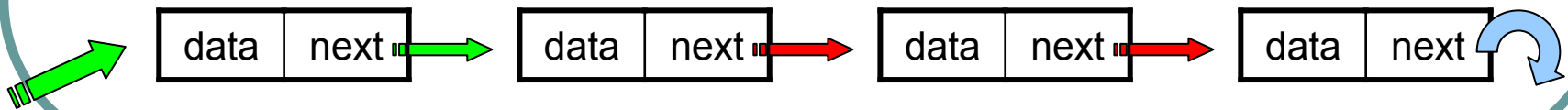


Využití spojených seznamů

- Seznam prvků
 - Přidání, mazání, vyhledávání, výpis prvků
- Zásobník (LIFO)
 - Přidání (na vrchol zásobníku), odebrání (z vrcholu zásobníku), je zásobník prázdný?
- Fronta (FIFO)
 - Přidání prvku, odebrání prvku, je fronta prázdná

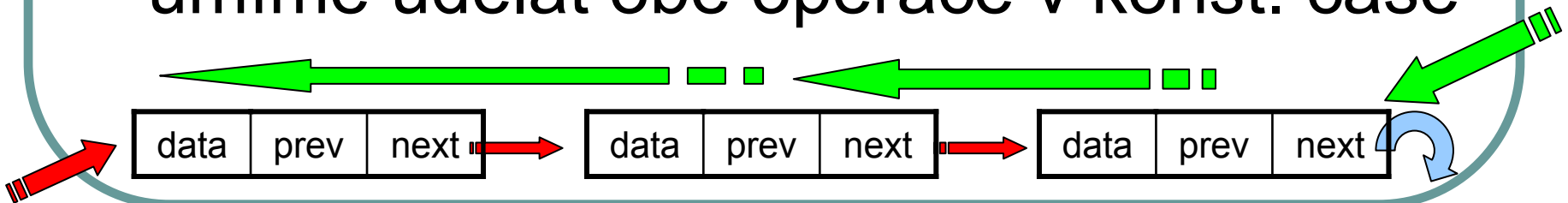
Zásobník (LIFO)

- Stačí jednosměrně zřetězený seznam
- Přidáváme na začátek seznamu
- Odebíráme ze začátku seznamu
 - Pokud je seznam neprázdný
- Obě operace v konstantním čase



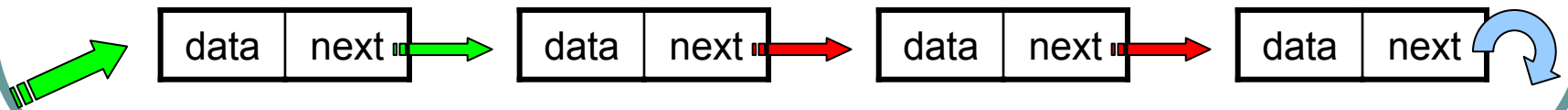
Fronta (FIFO)

- Můžeme použít jednosměrně i obousměrně zřetězený seznam
- Přidáváme na začátek a odebíráme z konce
 - Nebo obráceně
- U oboustranně zřetězeného seznamu umíme udělat obě operace v konst. čase



Fronta (FIFO)

- U jednostranně zřetězeného seznamu umíme buď konstantní přidání nebo konstantní odebrání
 - Druhá operace vyžaduje lineární procházení všech záznamů



Příští přednáška



- Následuje cvičení v B311 v 14:00
- Příští přednáška 13.10.2009
v B410 od 12:00