

Laboratoria nr 10

Lista jedno- i dwukierunkowa

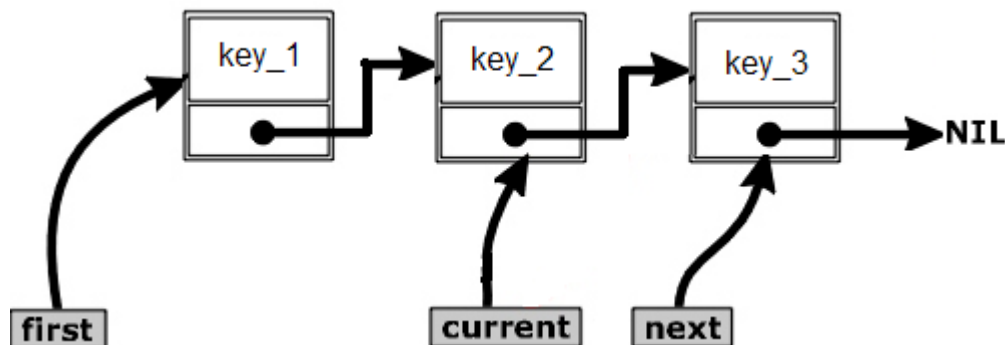
Materiały

Wyróżniamy następujące dynamiczne struktury danych:

- **Lista**

Lista jest liniowo uporządkowaną strukturą zawierającą zbiór elementów, z których dowolny element można usunąć oraz dodać w dowolnym miejscu. Pierwszy i ostatni element listy nazywamy *końcami listy*. Każdy element ma co najwyżej jednego następnika, który składa się z następujących składowych: klucza identyfikacyjnego, wskaźnika do następnika oraz dodatkowo dalszych informacji. Listy mogą być posortowane w porządku niemalejącym (w korzeniu znajduje się element o najmniejszej wartości klucza) lub nierosnącym (w korzeniu znajduje się element o największej wartości klucza). Rozważmy przypadek **jednokierunkowej listy posortowanej** w porządku niemalejącym. Aby dodać nowy element należy sprawdzić, w którym miejscu powinien się on znajdować. Dokonywane jest odpowiednie sprawdzenie rozpoczynające się w korzeniu i przechodzenie przez kolejne elementy w liście, tak długo dopóki nowy element, który chcemy dodać jest większy od badanego węzła i mniejszy od jego następnika, wtedy należy umieścić go między nimi. Należy, więc ustawić wskaźnik aktualnego węzła na dodawany element, a wskaźnik tego elementu na następnika. Ponieważ jest to lista jednokierunkowa, przeszukiwanie jej należy zawsze zaczynać od korzenia. Dodając, więc pierwszy element do pustej listy należy zapamiętać jego wskaźnik, by później mieć dostęp do tej struktury dzięki niemu. Listy mogą zawierać dwa wskaźniki. Takie listy nazywamy **dwukierunkowymi**, które pozwalają na poruszanie się w niej w obydwu kierunkach, co przyspiesza wszystkie operacje.

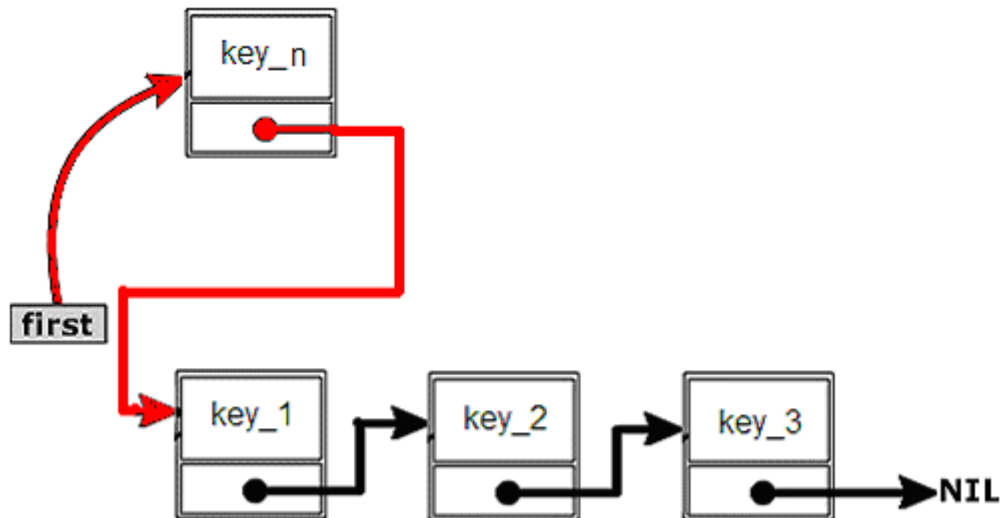
- *Jednokierunkowa*



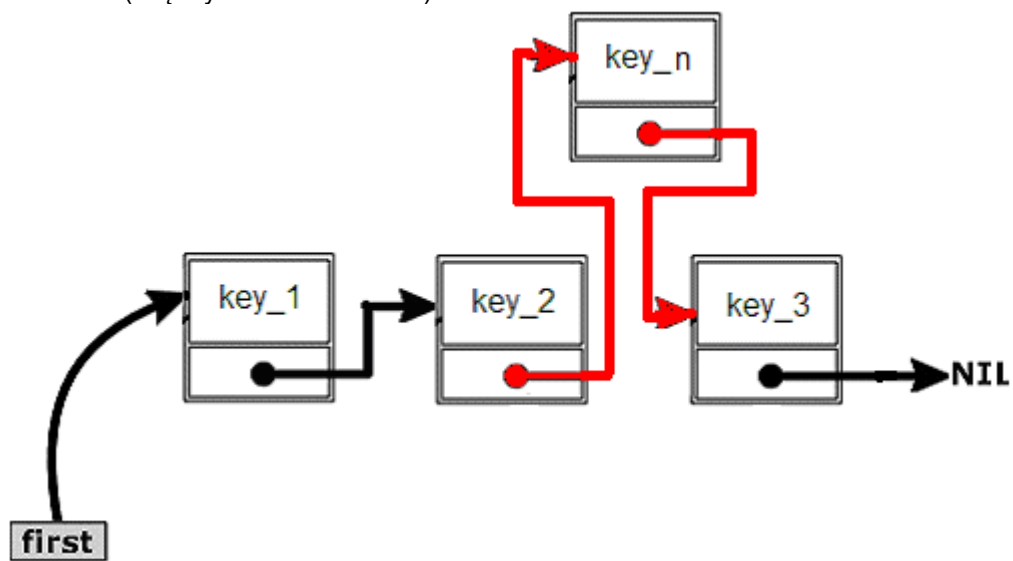
Przeszukiwanie dokonuje się od pierwszego elementu na liście (wskazywanego przez wskaźnik *first*) do ostatniego, który w polu swojego następnika wskazuje na wartość pustą (*NIL*).

Dodawanie nowych elementów

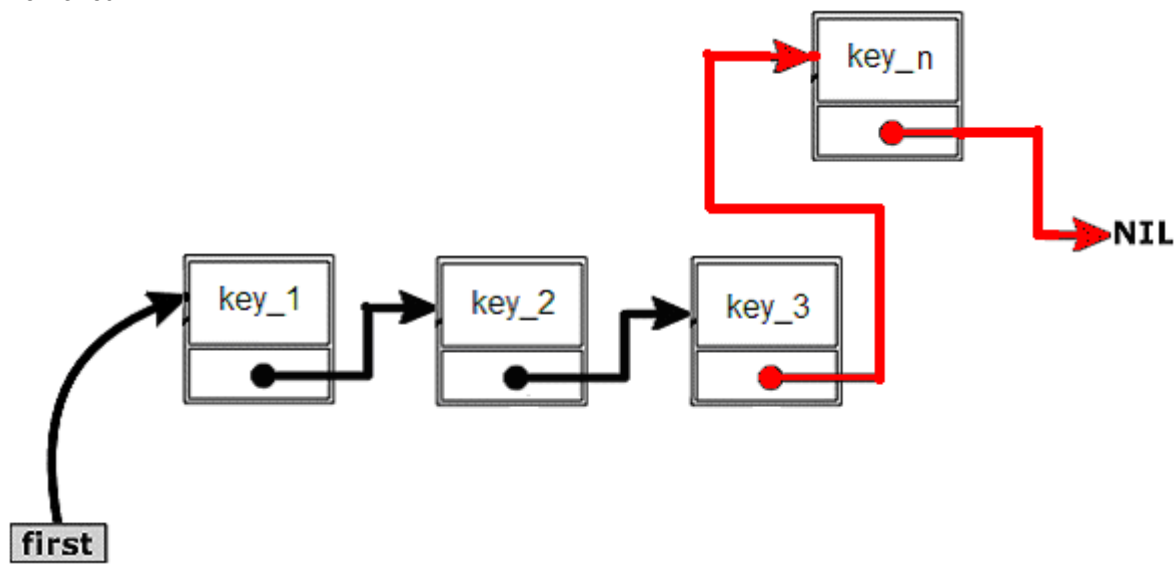
a) na początku



b) w środku (między 2 a 3 elementem)

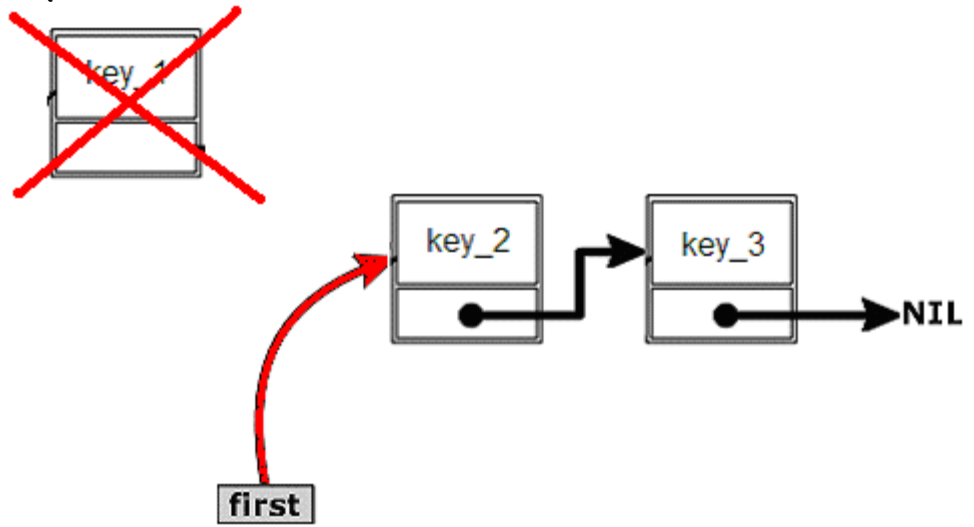


c) na końcu

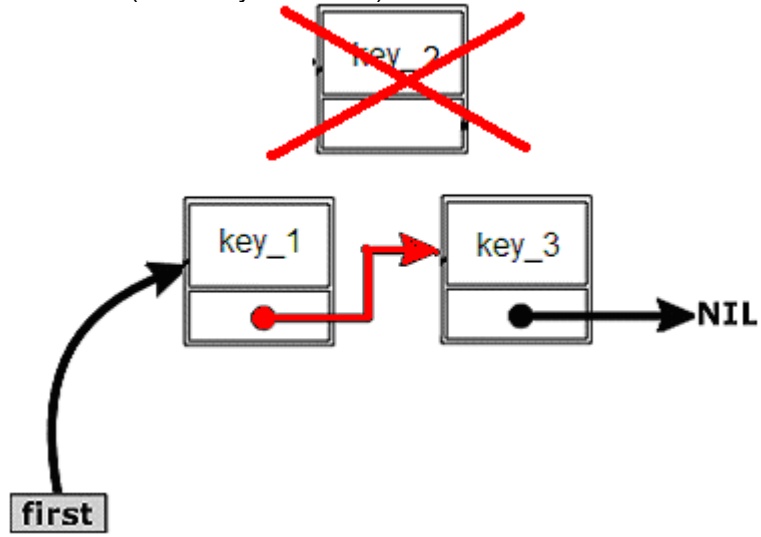


Usuwanie istniejących elementów:

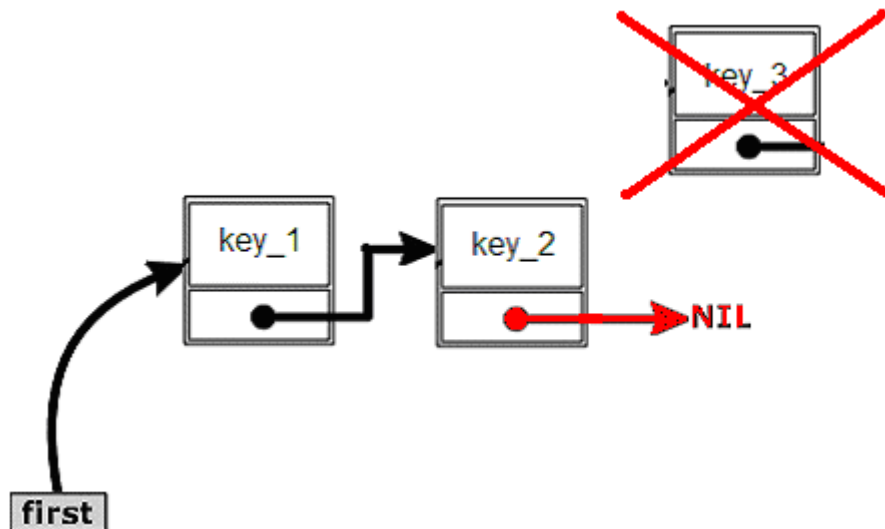
a) z początku



b) ze środka (usuwamy 2 element)



c) z końca

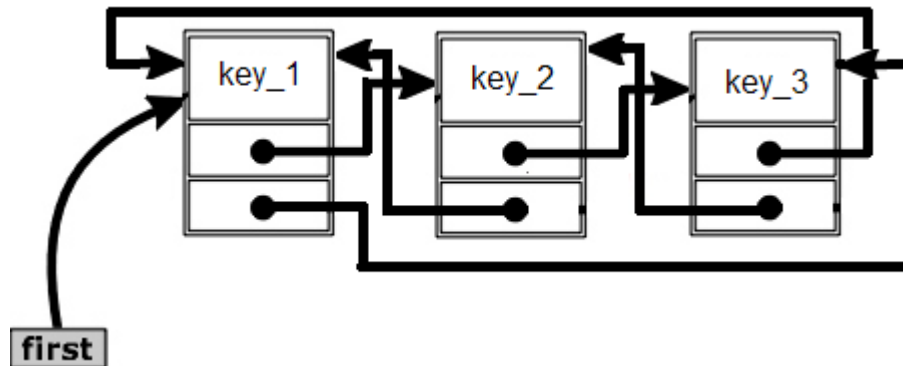


Przykładowa implementacja rekordu listy jednokierunkowej

```
struct lista{  
    int key;  
    struct lista *next;  
};
```

o Dwukierunkowa

Lista dwukierunkowa różni się od listy jednokierunkowej tylko tym, że każdy element w swojej strukturze oprócz klucza identyfikacyjnego posiada dwa wskaźniki *next* (wskazuje na następny element na liście) i *previous* (wskazuje na poprzedni element na liście) – dla porównania lista jednokierunkowa posiada tylko wskaźnik *next*.



Na liście dwukierunkowej można wykonywać operacje dodawania, usuwania elementów oraz przeszukiwania tej struktury w sposób analogiczny jak w przypadku listy jednokierunkowej. Trzeba tylko zwrócić baczną uwagę na fakt **zapewnienia spójności podczas dodawania lub usuwania elementów dla dwóch wskaźników**, a nie jak w przypadku listy jednokierunkowej dla jednego. Posiadanie dwóch wskaźników przez każdy element na liście powoduje, że brak ograniczenia związanego ze sposobem przeszukiwania listy (tylko od korzenia do liścia w przypadku listy jednokierunkowej), więc można się przesuwać po liście w obu kierunkach i z dowolnego miejsca, którego wskaźnik jest znany. Powyższy fakt powoduje, że struktura ta jest znacznie efektywniejsza pod względem przeszukiwania, ale niestety więcej czasu potrzebne jest na zarządzanie nowymi elementami. W formie ćwiczeń należy spróbować przerysować powyższe diagramy w taki sposób, aby odpowiadały wykonywanym operacjom dla listy dwukierunkowej.

Przykładowa implementacja rekordu listy dwukierunkowej

```
struct lista{  
    int key;  
    struct lista *next;  
    struct lista *prev;  
};
```