

# Podstawy Programowania

## Zajęcia laboratoryjne 12

### I rok Bioinformatyki Politechniki Poznańskiej

#### Lista dwukierunkowa

## 1 Lista dwukierunkowa

**Lista dwukierunkowa** w odróżnieniu od listy jednokierunkowej w swojej strukturze oprócz danych, które zawiera, zawiera także dwa wskaźniki. Jeden z nich wskazuje na następny element, a drugi na poprzedni element. Dzięki takiej konstrukcji możemy się w liście poruszać w dwóch kierunkach (co nie jest możliwe w przypadku listy jednokierunkowej). Powyższy fakt powoduje, że struktura ta jest znacznie efektywniejsza pod względem przeszukiwania, ale niestety więcej czasu potrzebne jest na zarządzanie nowymi elementami.

**Lista dwukierunkowa - schemat poglądowy:**

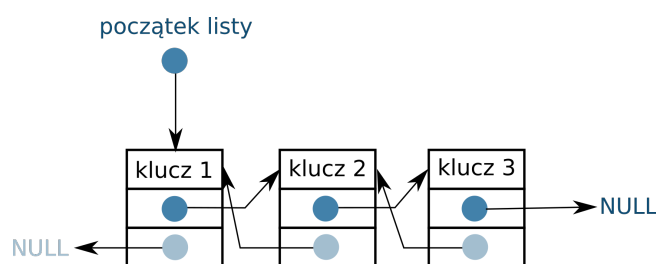


Figure 1: Schemat listy dwukierunkowej.

**Deklaracja listy dwukierunkowej**, w poniższym przykładzie zdefiniowano listę dwukierunkową, która zawiera następujące składowe: klucz identyfikacyjny, wskaźnik na następny węzeł (adres do następnika) i wskaźnik na poprzedni węzeł (adres do poprzednika):

---

```
1 //definicja wezla na liscie: klucz, wskaźnik na następnik i poprzednik
2 struct lista
3 {
4     int klucz;
5     struct lista *next;
6     struct lista *prev;
7 };
```

---

**Dodawanie nowego elementu**, jest podstawową operacją na liście. Elementy mogą być dodawane na początek listy, za wskazanym elementem lub na koniec listy. Podobnie jak w przypadku listy jednokierunkowej. Należy jednak pamiętać o zachowaniu spójności struktury dla dwóch wskaźników, a nie tylko jednego. Zwróć uwagę na przedstawione schematy.

**Dodaj nowy element na początek listy:**

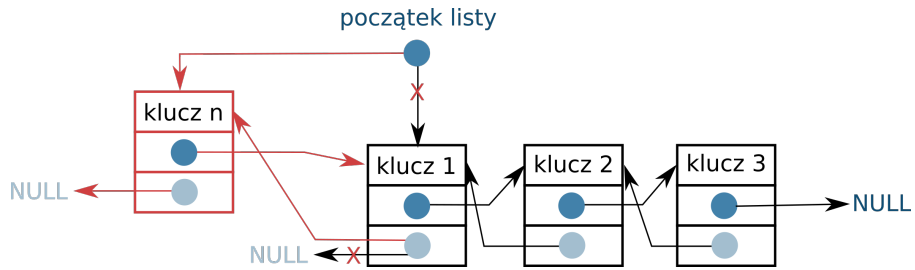


Figure 2: Dodawanie nowego elementu na początek listy.

**Dodaj nowy element na koniec listy:**

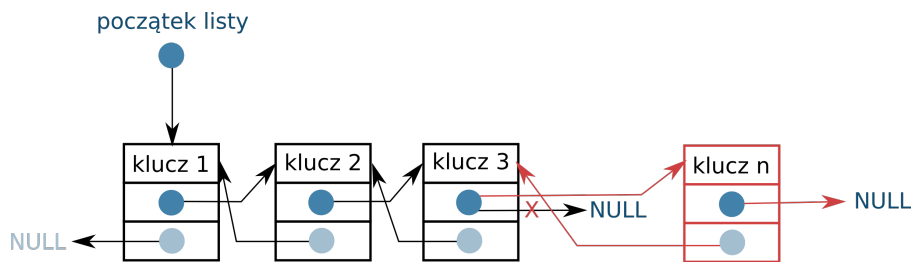


Figure 3: Dodawanie nowego elementu na koniec listy.

**Usuwanie elementu**, jest kolejną podstawową operacją na listach. Podobnie jak przy dodawaniu elementu, usuwanie też można wykonać na kilka sposobów, usuwać możemy z początku listy, z końca listy, po wskazanej wartości klucza, czy po konkretnym indeksie (zależne od zdefiniowania zadania). Niezależnie od wyboru opcji usuwania należy odpowiednio ustawić wskaźniki, tak aby zachować ciągłość struktury, pamiętaj że w przypadku listy dwukierunkowej mamy nie tylko wskaźnik na następny element, ale także na element poprzedni. Zwróć uwagę na różnice przedstawione na poniższych schematach:

**Usuń element z początku listy:**

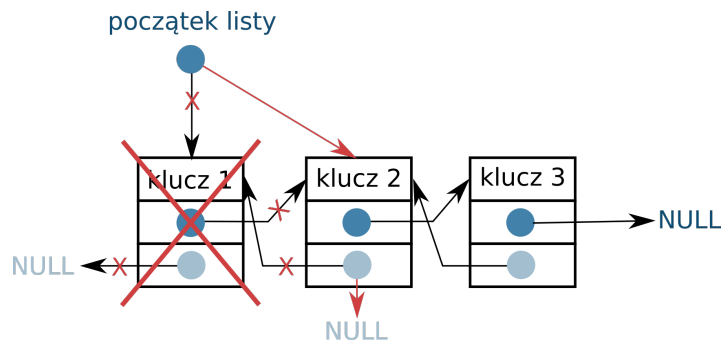


Figure 4: Usuwanie elementu z początku listy.

Usuń element z końca listy:

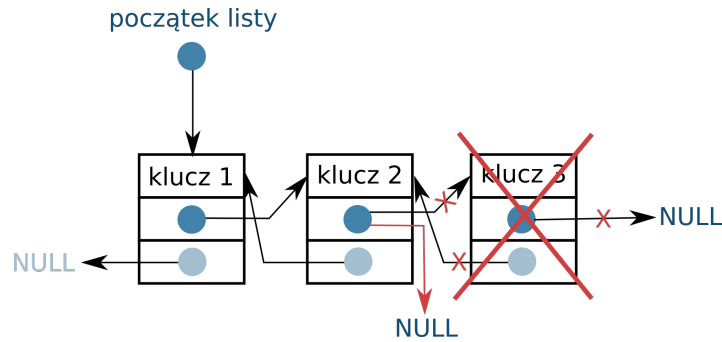


Figure 5: Usuwanie elementu z końca listy.

Aby mieć pewność, że rozumiesz jak działają listy dwukierunkowe, zastanów się jak wyglądałyby schematy dodawania nowego elementu za wskazany element w liście dwukierunkowej i usuwanie dowolnego elementu z listy dwukierunkowej. Następnie przerób listę jednokierunkową na listę dwukierunkową (wykorzystaj kod umieszczony na stronie w ramach poprzednich zajęć: lab 11).

**Zad 1.** Przerób funkcję dodawania z listy jednokierunkowej na listę dwukierunkową, tak aby dodawała wartość na początek listy, jeśli lista jest pusta. W innym przypadku dodawaj wartość na koniec listy.

**Zad 2.** Przerób funkcję usuwania z listy jednokierunkowej na listę dwukierunkową, tak aby funkcja usuwała wartości z początku listy.

**Zad 3.** Przerób funkcję dodawania nowego elementu za wskazanym elementem z listy jednokierunkowej na listę dwukierunkową, tak aby dodawała wartość za wskazanym elementem listy.

## 2 Zadania

Zad 1. Zaprojektuj listę dwukierunkową służącą do przechowywania liczb całkowitych. Stwórz program umożliwiający obsługę takiej listy: możliwość dodawania, usuwania, modyfikowania elementów, wyszukiwania elementów listy (dowolnego, zadanego przez użytkownika), wyświetlania, (opcjonalnie sortowania całej listy) Funkcje dodawania i usuwania powinny być dwie: dodaj przed wskazanym elementem i dodaj za wskazanym elementem listy oraz usuń przed wskazanym elementem i usuń po wskazanym elemencie.

Zad 2. Wykorzystaj zadanie 2 z lab 11 i przerób kod na listę dwukierunkową.