# 4- Spojové struktury

- Struktura obsahující soubor záznamů propojených pomocí referencí
- Spoj vyjadřuje vztah předchůdce-následník
- Ve zřetězené struktuře musí mít každý prvek odkaz na další prvek

## Spojový seznam

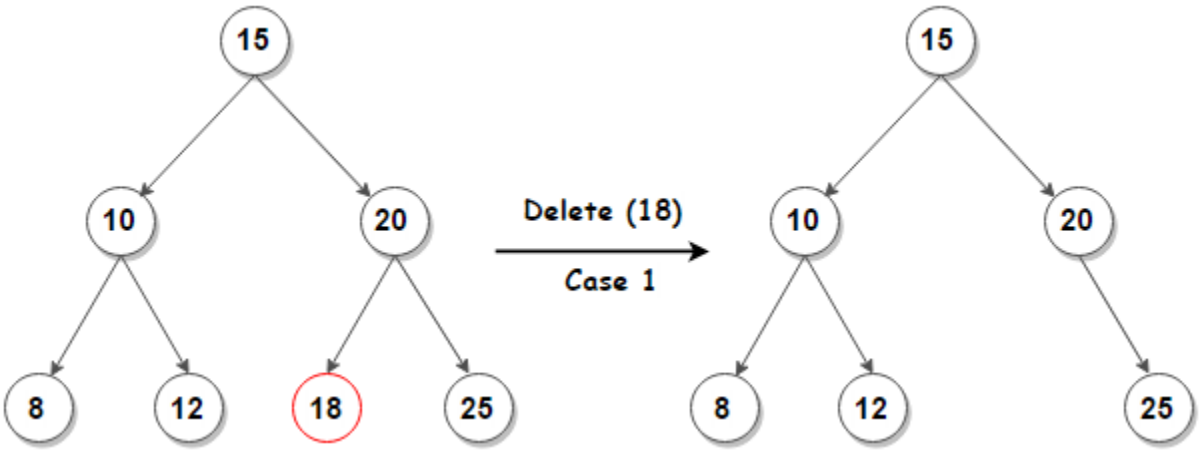
- Struktura realizující seznam dynamické délky
- Každý prvek seznamu obsahuje datovou část a odkaz na další prvek v seznamu
- Může být i obousměrný nebo kruhový
- Rychlý, nezabírá mnoho místa, není omezen délkou, jednoduché vkládání doprostřed seznamu
- Nemožnost přístupu na index (je potřeba jít postupně)
- Použití při implementaci Stack a Queue
-

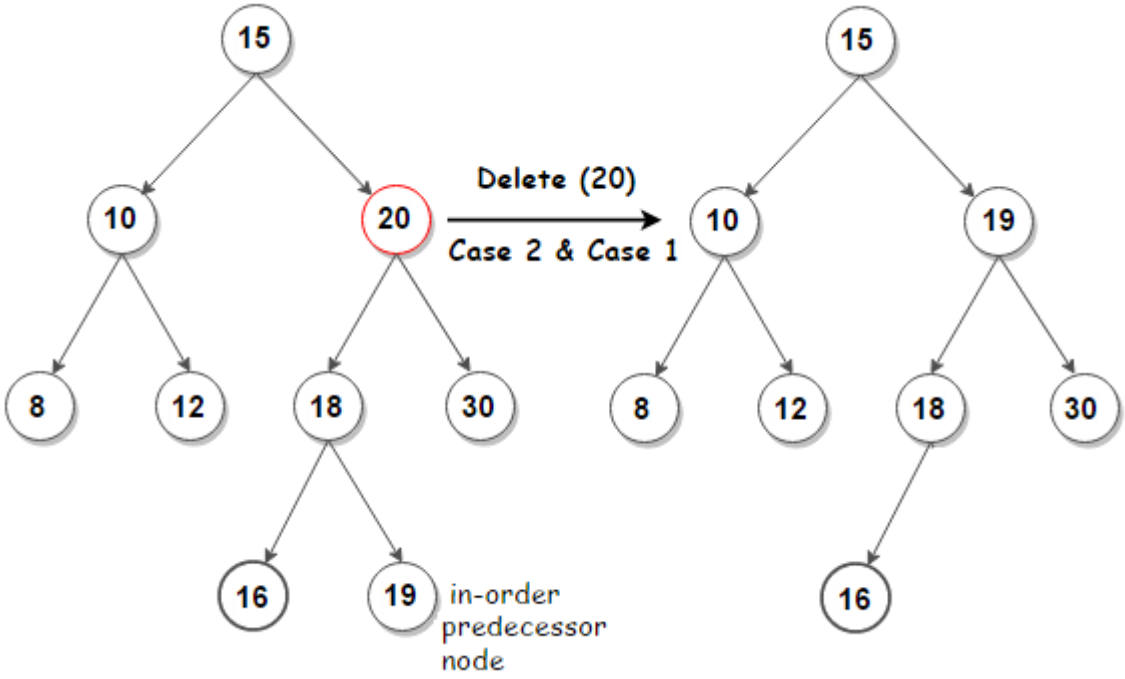
## Spojové stromy

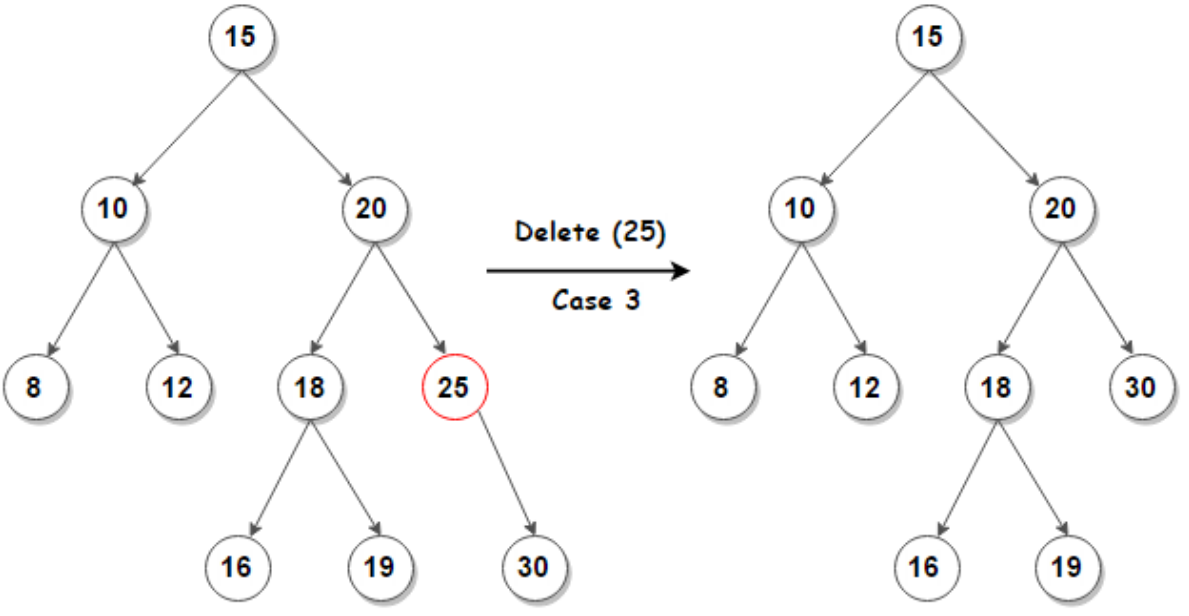
- Každý může mít následníky
- Striktní Binární strom: Každý prvek má dva nebo žádné potomky
- Plně binární strom: Každý prvek má dva potomky na stejné úrovni
- Binární strom má v levé větvi menší prvky než kořen a v pravé větší prvky
- Hledání a řazení dat
- Procházení stromu:

```
0 references
public void VypisObsah()
{
    if (koren != null)
    {
        VypisObsahRekur(koren);
    }
}
3 references
private void VypisObsahRekur(Node koren)
{

    if (koren.levy != null)
    {
        VypisObsahRekur(koren.levy);
    }
    Console.WriteLine(koren.hodnota);


    if (koren.pravy != null)
    {
        VypisObsahRekur(koren.pravy);
    }

}
```

- Mazání ze stromu:
  - Pokud mažeme list, stačí pouze smazat odkaz na něj
  - Pokud má jednu větev, je vynechán
  - Pokud má oba následníky, najdeme nejbližší prvek (nejnižší nebo největší) a ten dosadíme na místo smazaného prvku



## Fronta

- FIFO kolekce (first in first out)
- Přidávání prvku na začátek fronty (queue)
- Odebírání z konce fronty (enqeue)

## Zásobník

- LIFO (last in first out)
- Přidání (push)
- Mazání (pop)