

Problemstillingen

- Implementere datastrukturer i C++
 - Dynamic Arrays
 - Linked lists
 - Linked Array List

Temaer i IN1910

- C++
- Objektorientering
- Datastrukturer
- Kompleksitet, Big Oh
- Gradvis utvikling, lesbarhet
- Testing

Løsning

```
1 void LinkedList::insert (int val, int idx)
2 {
3     if ((idx < 0) || (idx > (int) length()))
4         throw std::range_error(
5             "Insert index " + std::to_string(idx)
6             + " out of range for array of length " + std::to_string(length())
7         );
8     else if (idx == (int) length())
9         append(val);
10    else if (idx == 0){
11        Node *new_node = new Node;
12        new_node->value = val;
13        new_node->next = _head;
14        _head = new_node;
15        _length++;
16    } else {
17        Node *node_before = _head;
18        for (int i=0; i<idx - 1; i++)
19            node_before = node_before->next;
20        Node *node_after = node_before->next;
21
22        Node *new_node = new Node;
23        new_node->value = val;
24        new_node->previous = node_before;
25        new_node->next = node_after;
26
27        node_before->next = new_node;
28        node_after->previous = new_node;
29        _length++;
30    }
31 }
```

```
1 int LinkedList::pop (int idx)
2 {
3     const int val = operator[](idx);
4     remove(idx);
5     return val;
6 }
```

```
1 LinkedList::LinkedList (std::vector<int> v) : _length(0)
2 {
3     for (auto e : v)
4         append(e);
5 }
6
7
8 LinkedList::~~LinkedList ()
9 {
10     const unsigned int num_nodes_to_dealloc = _length;
11     for (size_t i=0; i<num_nodes_to_dealloc; i++)
12         remove(0);
13 }
```

Utfordringer

- Ta hensyn til alle edge cases i metodene
 - Gjenbruke metoder i steder for å implementere ny logikk, selv om effektivitet går ned
- Pointer-syntaks
 - Holde tunga rett

Testing

- Unit tests
 - Funksjonaliteten vi ønsker fra klassen er tydelig definert
 - Vi har hele tiden full kontroll på hvordan den interne tilstanden skal være – lett å teste!