To split off 2nd half of a list:

1. **O(1)** create a new list
2. **O(n)** for i = 1 to \( \left\lfloor \frac{n}{2} \right\rfloor \)
3. **O(1)** remove the last item
4. **O(1)** add it to the front of the new list

Efficiency - depends on implementation:

- **O(n)** total for doubly-linked list
- **O(n^2)** for singly-linked list

Diagram:

```
12345678
```

```python
list = [1, 11, 23, 16, 49, 2]
new_list = [2]
for i in range(1, len(list) // 2 + 1):
    last_item = list.pop()
    new_list.insert(0, last_item)
```

```
new_list = [2, 16, 49, 2]
```
List Operations

doubly-linked list

add to front \( O(1) \) \( O(n) \)
add to end \( O(1) \) \( O(1) \) - typical case (no resize)
\( O(n) \) - worst case (resize)
\( O(1) \) - amortized

remove from front \( O(1) \) \( O(n) \)
remove from end \( O(1) \) \( O(1) \)

get \( O(n) \) worst case \( O(1) \)