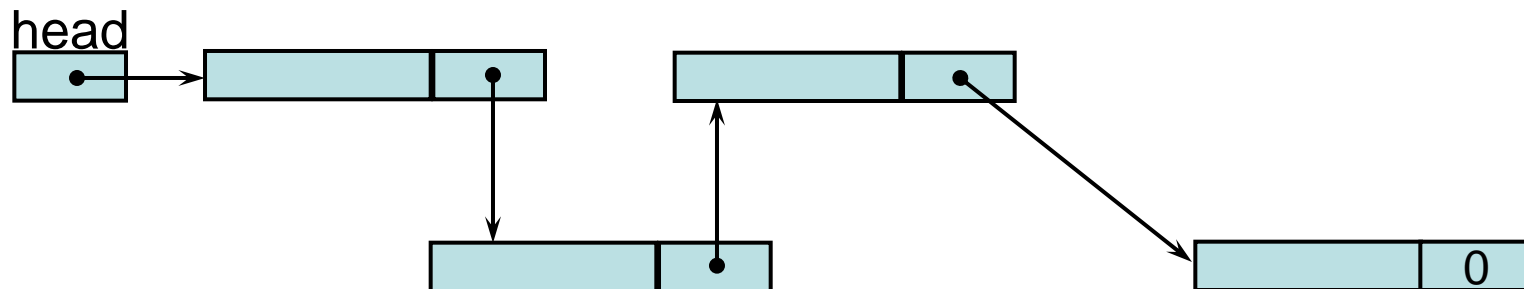


Queue - dynamisch - RemoveElement (1)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

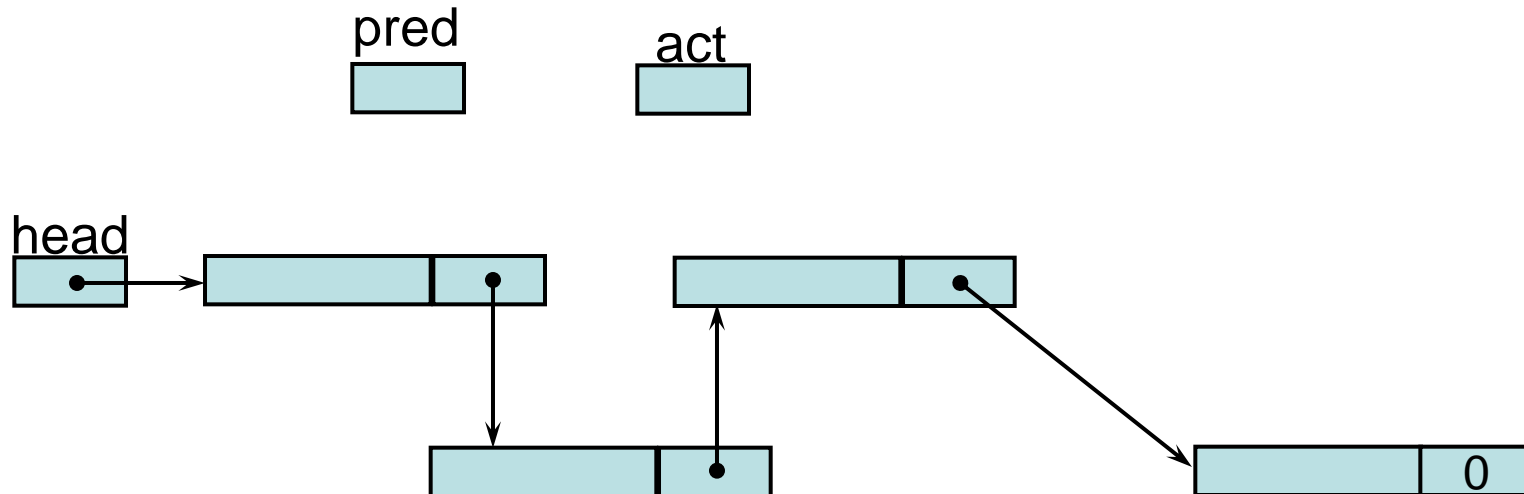


Queue - dynamisch - RemoveElement (2)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3



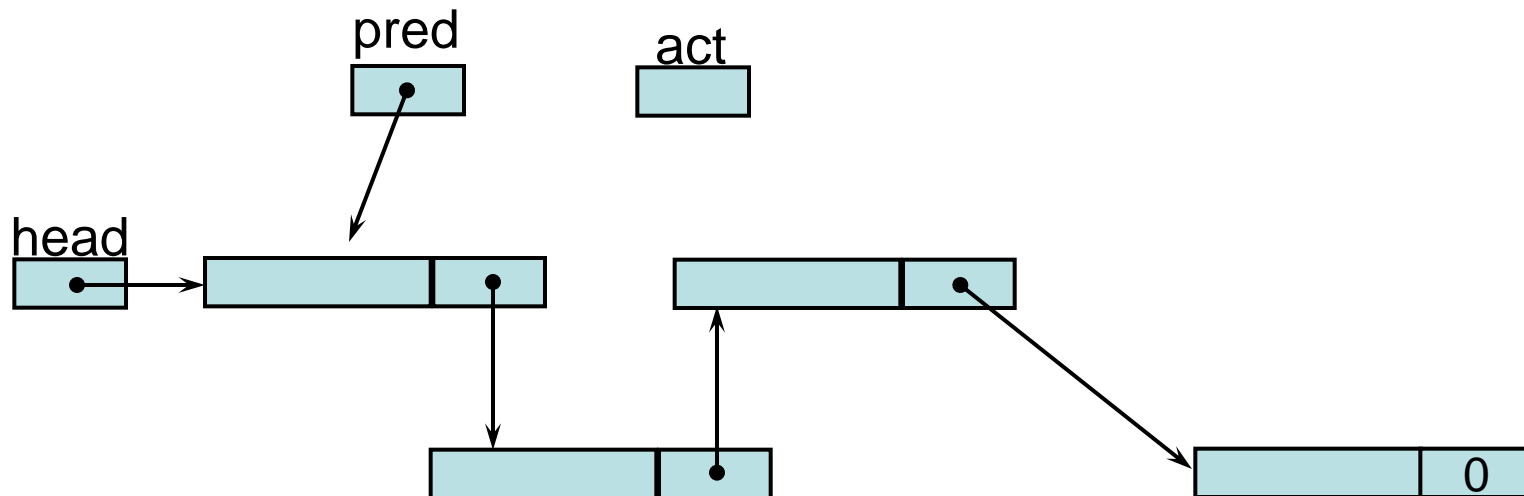
Queue - dynamisch - RemoveElement (3)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 2



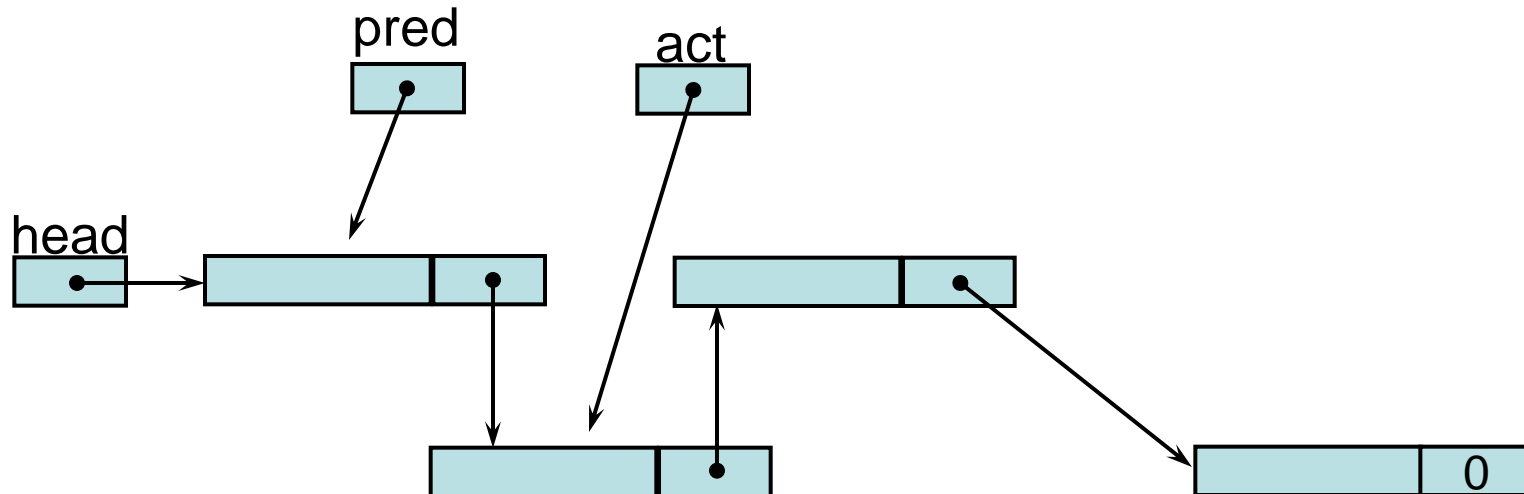
Queue - dynamisch - RemoveElement (4)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 2



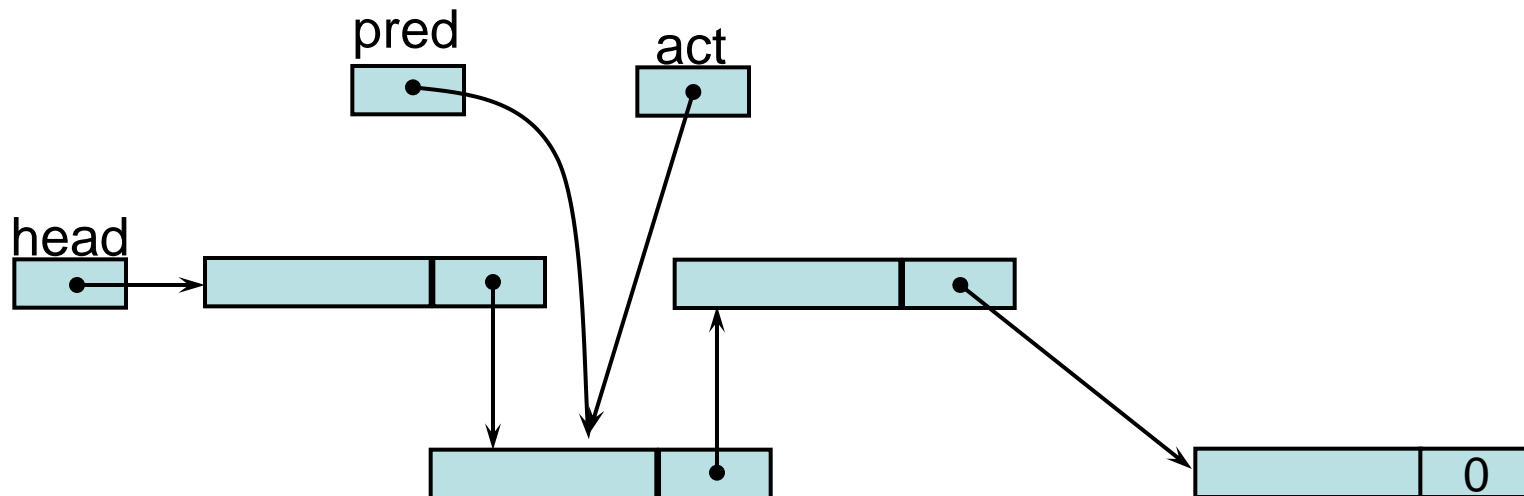
Queue - dynamisch - RemoveElement (5)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 2



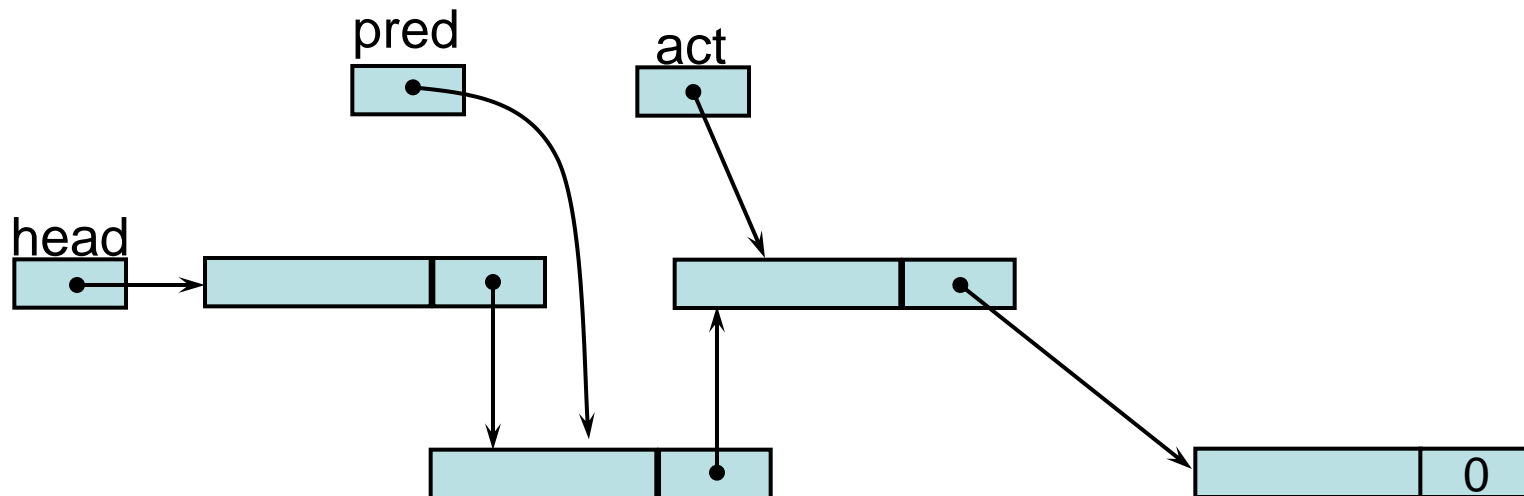
Queue - dynamisch - RemoveElement (6)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 3



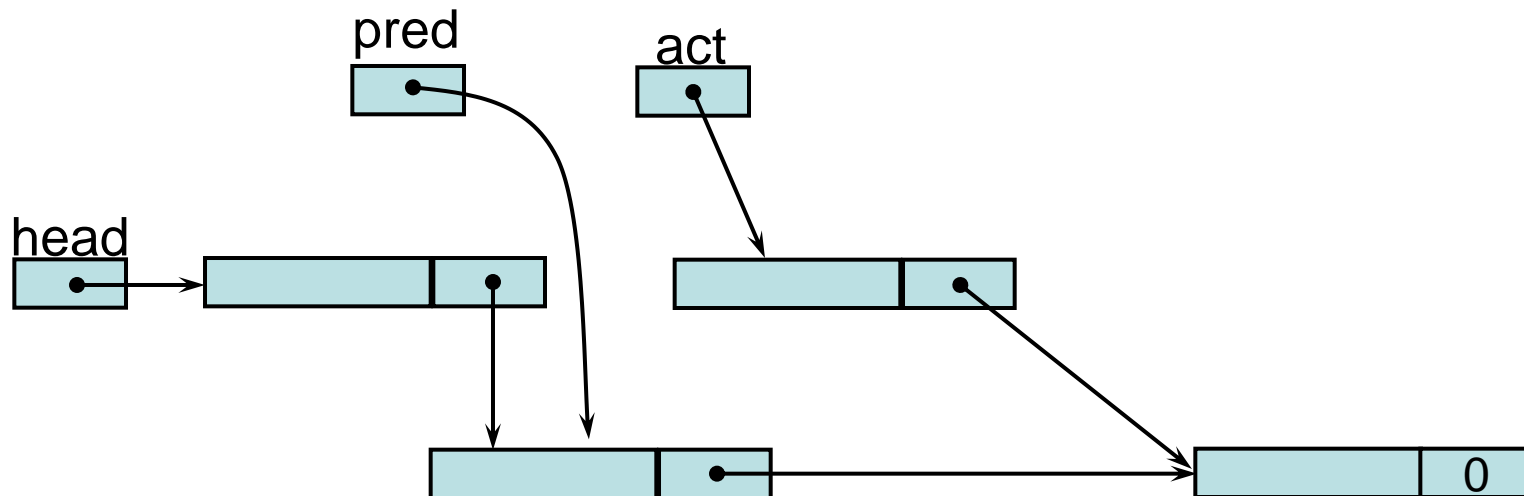
Queue - dynamisch - RemoveElement (7)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 3



Queue - dynamisch – RemoveElement (8)

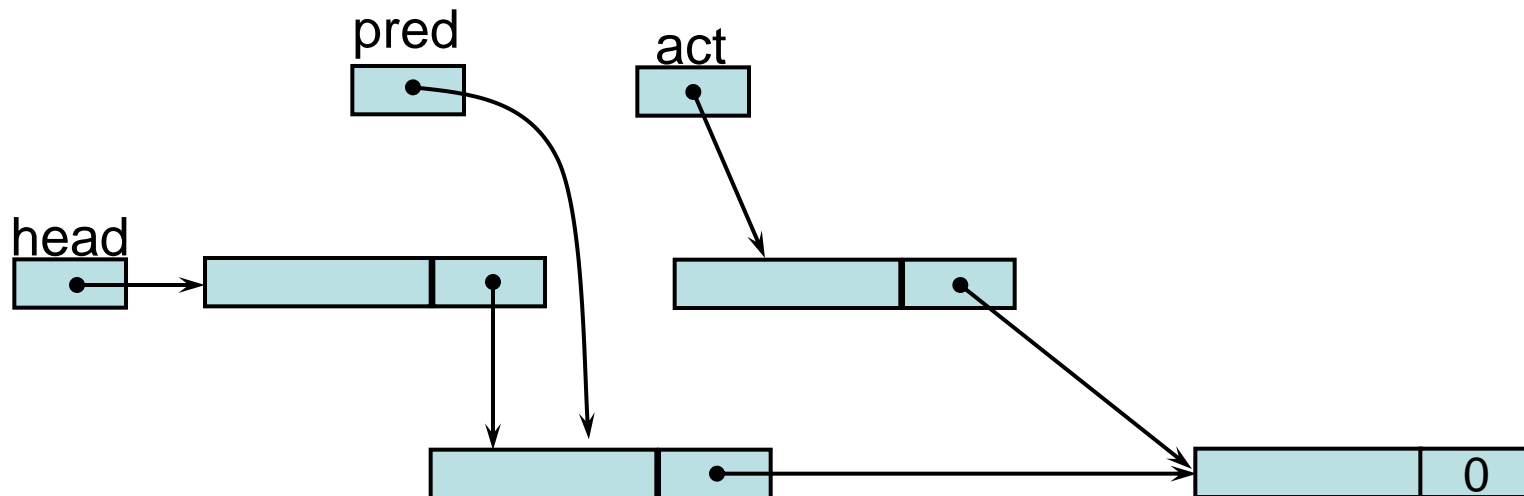


universität
wien

```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 3

actpos = 3



Queue - dynamisch - RemoveElement (9)



```
void List::RemoveElement(int pos) {  
    Element* pred, * act;  
    int actpos = 2;  
    if(pos == 1) RemoveFirst();  
    else {  
        pred = head;  
        act = head->next;  
        while(act != 0 && actpos < pos) {  
            pred = act;  
            act = act->next;  
            actpos++;  
        }  
        pred->next = act->next;  
        delete act;  
    }  
}
```

pos = 2

actpos = 3

