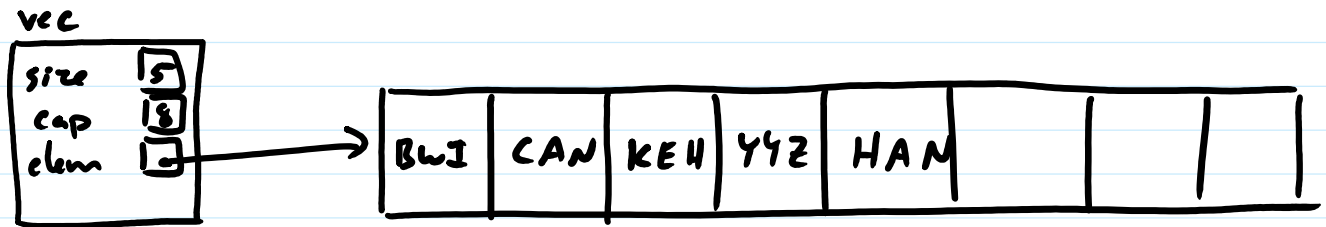
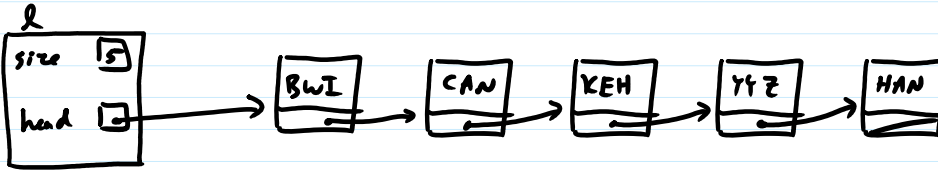


Midterm Exam LORIA 250



```
for (int i = 0; i < vec.size(); i++) // linear time overall O(n)
{
    process(vec[i]);
}
```

```
T& vector::operator[](int i)
{
    return elements[i]; // constant time O(1)
}
```



```

for (int i = 0; i < vec.size(); i++)
{
    process(l[i]); // 0 + 1 + 2 + ... + n-1 iterations of loop in operator[] O(n^2)
}

```

```

T& list::operator[](int i) // worst case O(n)
{
    node *curr = head;
    for (int j = 0; j < i; j++)
        curr = curr->next;
    return curr->data;
}

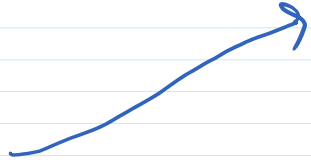
```

List ADT

⋮
 set iterator at beginning → begin
 set iterator at end → end

Iterator ADT

go to next item ++
 get current item *
 check whether finished != end()



```

for (CIVector<std::string>::iterator i = vec.begin(); i != vec.end(); i++) // overall O(n)
{
    process(*i);
}

```

```

class ListIterator
{
    item& operator*() { curr->data; }
    ListIterator& operator++(int) { curr = curr->next; return *this; }
private:
    node *curr;
}

```

```

for (int i = 0; i < l.size(); i++)
{
    get 1st thing from l
    do something with it
}

```

```

for (CIVector<std::string>::iterator i = vec.end(); i != vec.begin(); i--) // overall O(n)
{
    process(*i);
}

```

f