Pointers and Dynamic Arrays

Week 12
2017 Fall

Computer Programming for Engineers

Cassiano Campes <cassianocampes@gmail.com>
Yoohyuk Lim <dburg3065@gmail.com>

Problem 1: Polynomial Problem

Problem 1: Polynomial Problem

- Use dynamic arrays to implement a polynomial class

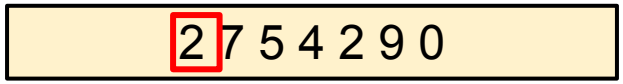
Class Polynomial

- **Constructors:**
  - Polynomial() as default constructor;
- **Private members:**
  - Index coefficient - integer value;
  - X value - pointer integer value;
- **Public members:**
  - Get/Set index coefficient;
  - Get/Set index value;

main() must have the variable that stores the value of X

Problem 1: Polynomial Problem

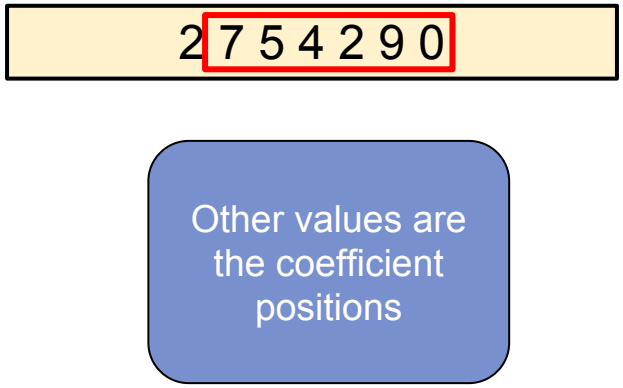
- We read a string with the values from the user:



First value indicates the X value of the polynomial

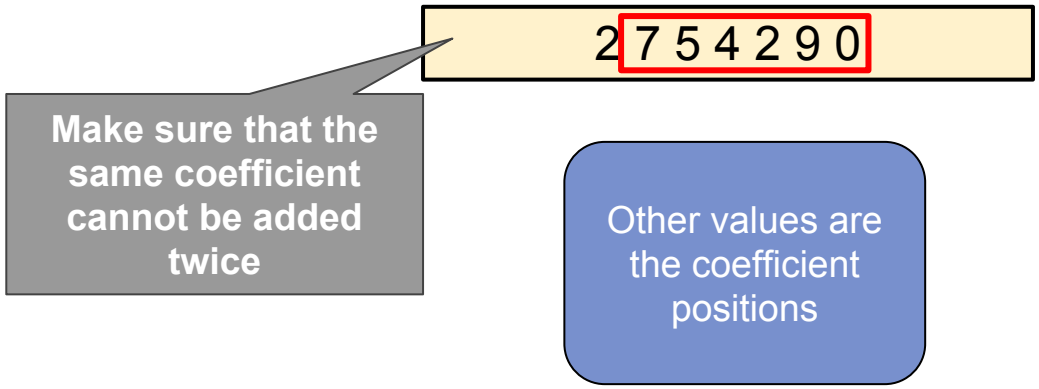
Problem 1: Polynomial Problem

- We read a string with the values from the user:



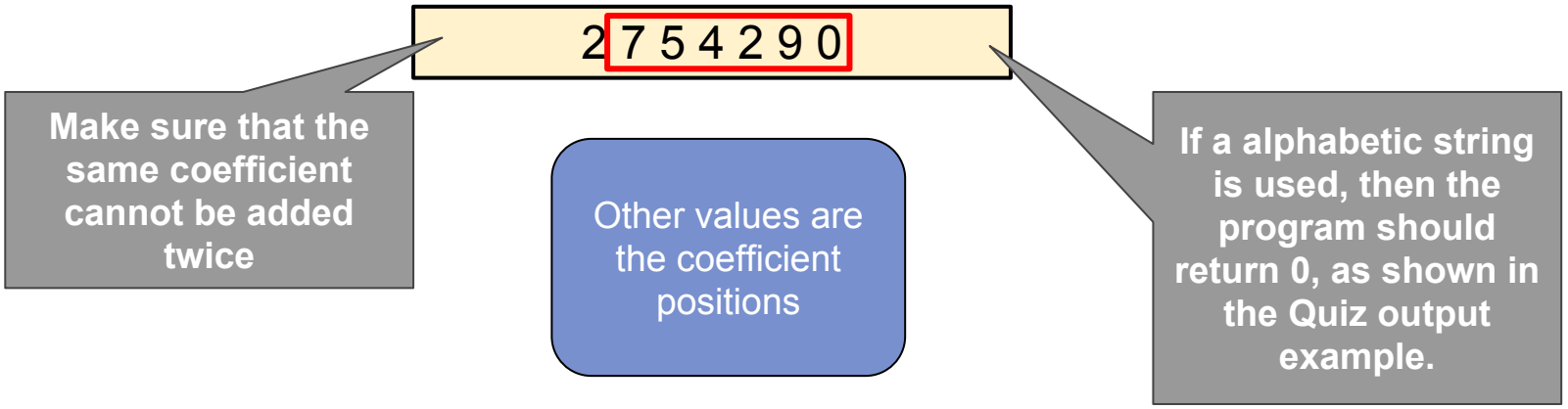
Problem 1: Polynomial Problem

- We read a string with the values from the user:



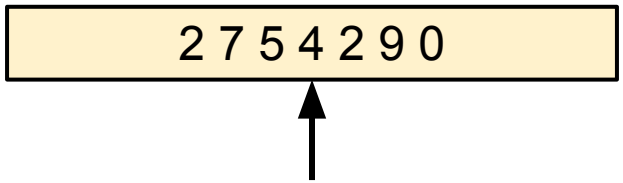
Problem 1: Polynomial Problem

- We read a string with the values from the user:



Problem 1: Polynomial Problem

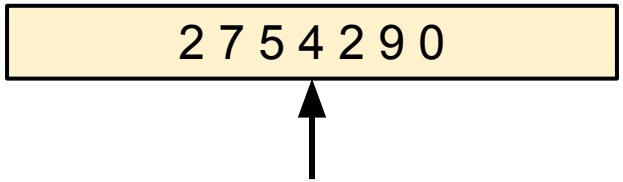
- We read a string with the values from the user:



$$f(x) = x^9 + x^7 + x^5 + x^4 + x^2 + x^0$$

Problem 1: Polynomial Problem

- We read a string with the values from the user:



$$f(2) = 2^9 + 2^7 + 2^5 + 2^4 + 2^2 + 2^0$$

Problem 1: Suggestions

1. Use getline(cin, input) to read the string from user
2. To get the individual values from the read input we can use **stringstream**:

```
string input;
getline(cin, input);
stringstream stream(input);
while(1) {
    int n;
    stream >> n;
    if(!stream) break;
   // … do whatever you want with 'n'
}
```

3. You could use vectors to store the read value temporarily in this example

Problem 1: Suggestions

4. After you have read the values, you must identify and remove the numbers that are repeated;
   a. If you used vector to store numbers temporarily, then:

```
int size = vec.size();
for (int i … ) {
    for (int j … ) {
        if (vec[j] is equal to vec[i])
            vec.erase(vec.begin() + j);
        if(size != vec.size()) {
            --j; size = vec.size();
            /* because we removed an entry, we need to update index */
        }
    }
}
```

Problem 1: Suggestions

4. After you have read the values, you must identify and remove the numbers that are repeated;
   a. If you used vector to store numbers temporarily, then:

```
int size = vec.size();
for (int i … ) {
    for (int j … ) {
        if (vec[j] is equal to vec[i])
            vec.erase(vec.begin() + j);
        if(size != vec.size()) {
            --j; size = vec.size();
            /* because we removed an entry, we need to update index */
        }
    }
}
```

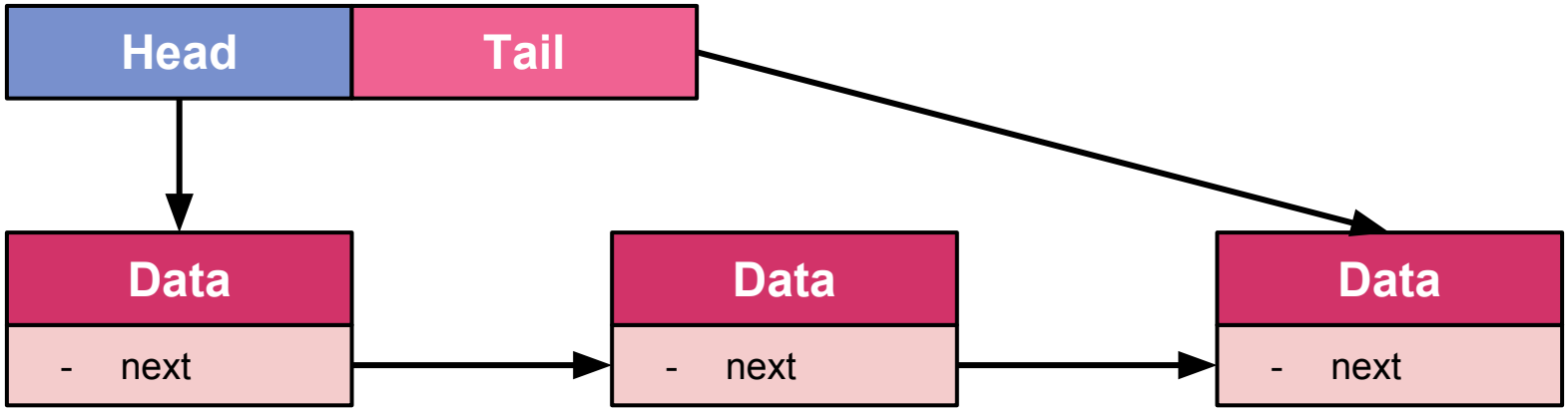
It erases the repeated number

5. Then you can create the dynamic array that stores the necessary number of polynomial;

Problem 2:
Polynomial Using Linked list

Problem 2: What are Linked Lists?

- Linked lists are one of the most famous data structures



Problem 2: Polynomial Using Linked list

- Let’s develop our number list using linked lists;

structure Polynomial

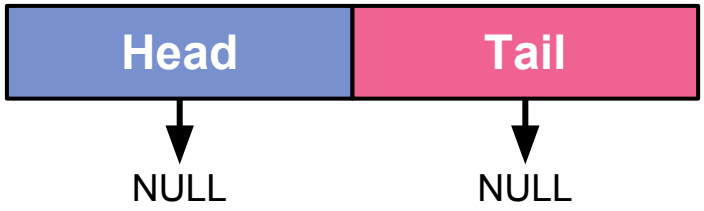
- Pointer to the X value;
- Integer to store the coefficient;
- Pointer to next structure Todo;

Class List()

- **Constructors:**
  - List() as default constructor;
- **Private members:**
  - Pointer to head and tail struct Polynomial;
- **Public members:**
  - Add an polynomial to the list;
  - Return the head pointer to the list;

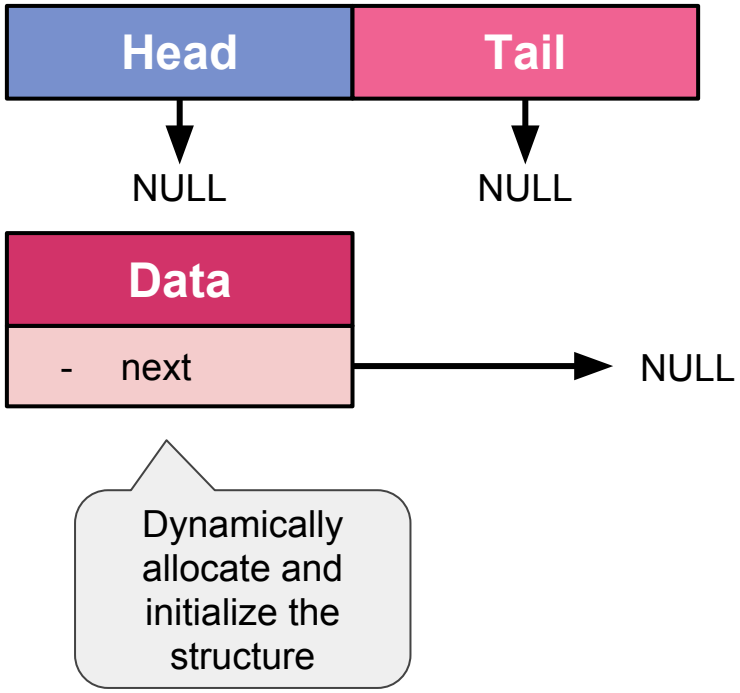
Problem 2: Suggestions

Empty list!



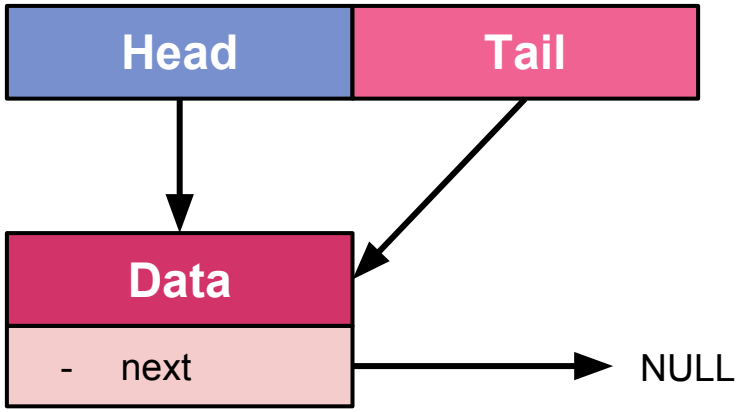
Problem 2: Suggestions

Adding an entry to the list!



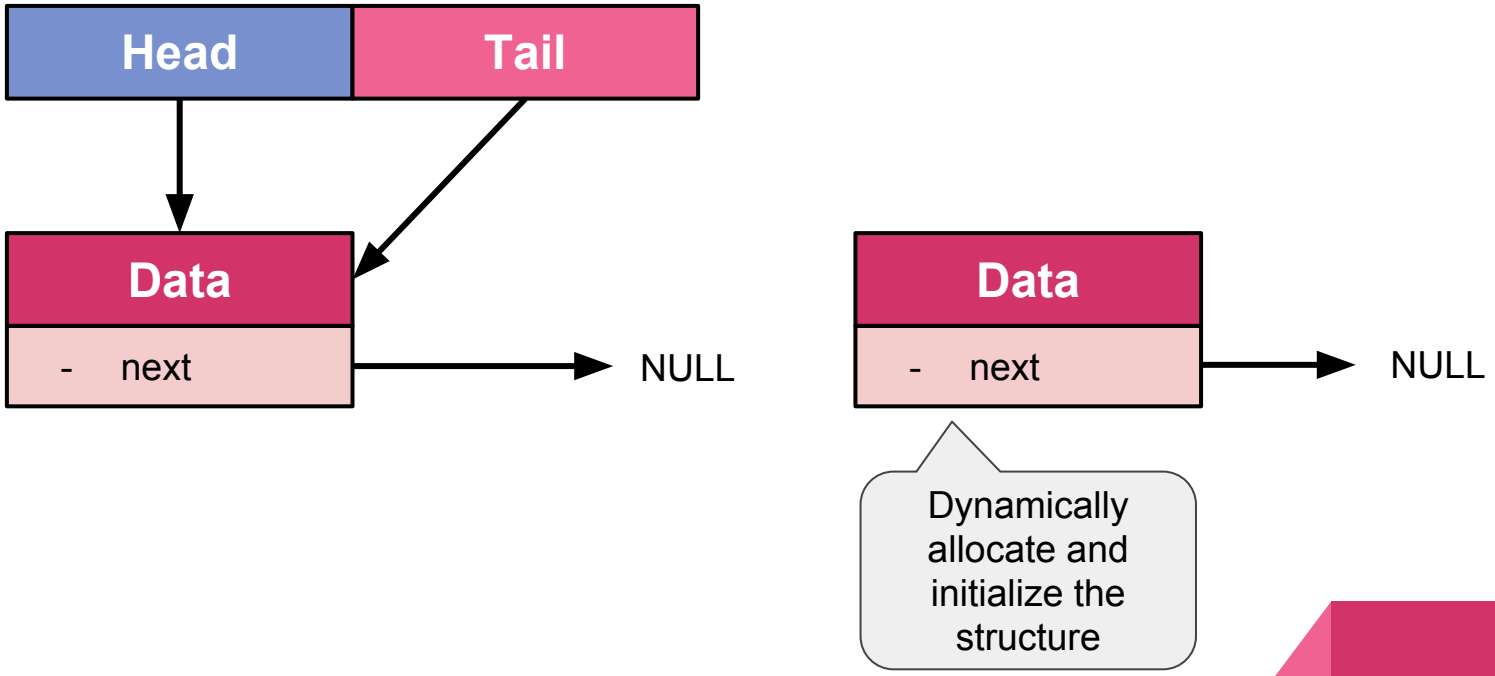
Problem 2: Suggestions

Adding an entry to the list!



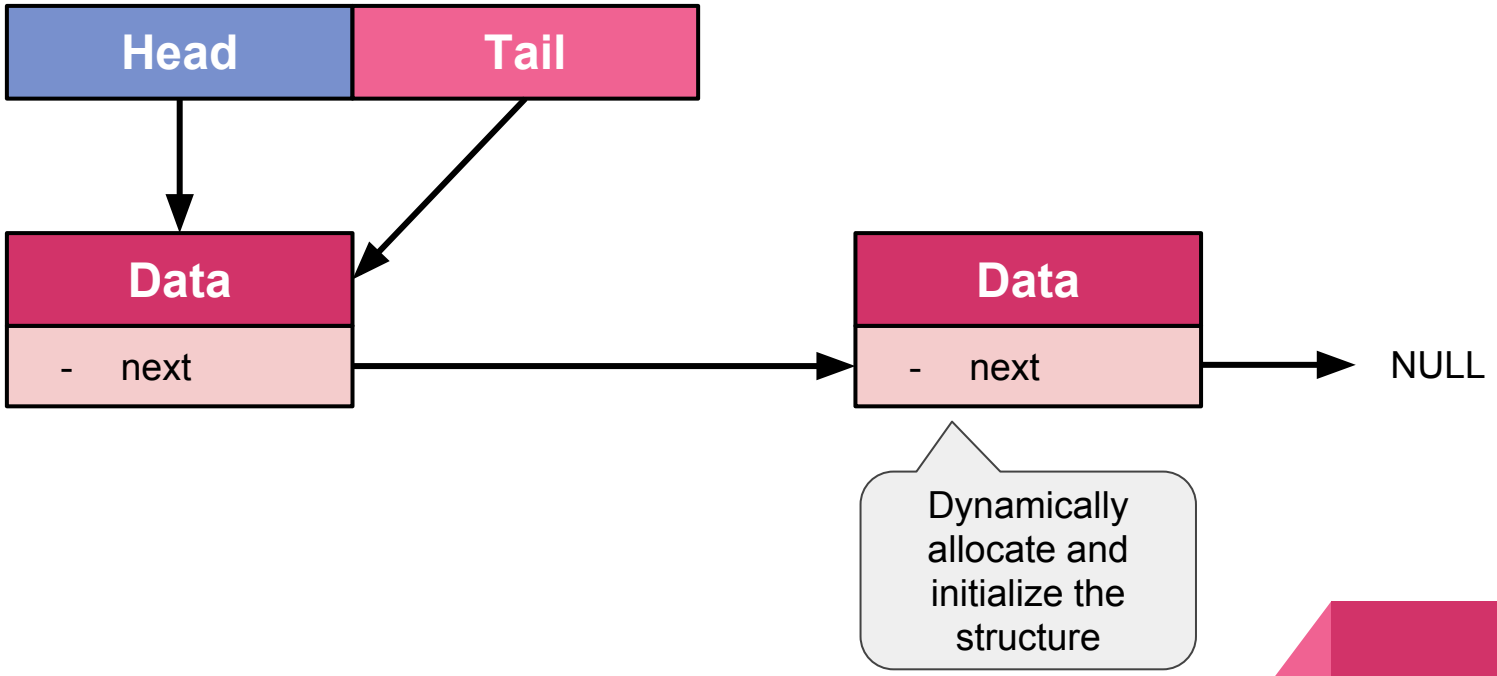
Problem 2: Suggestions

Adding an entry to the list!



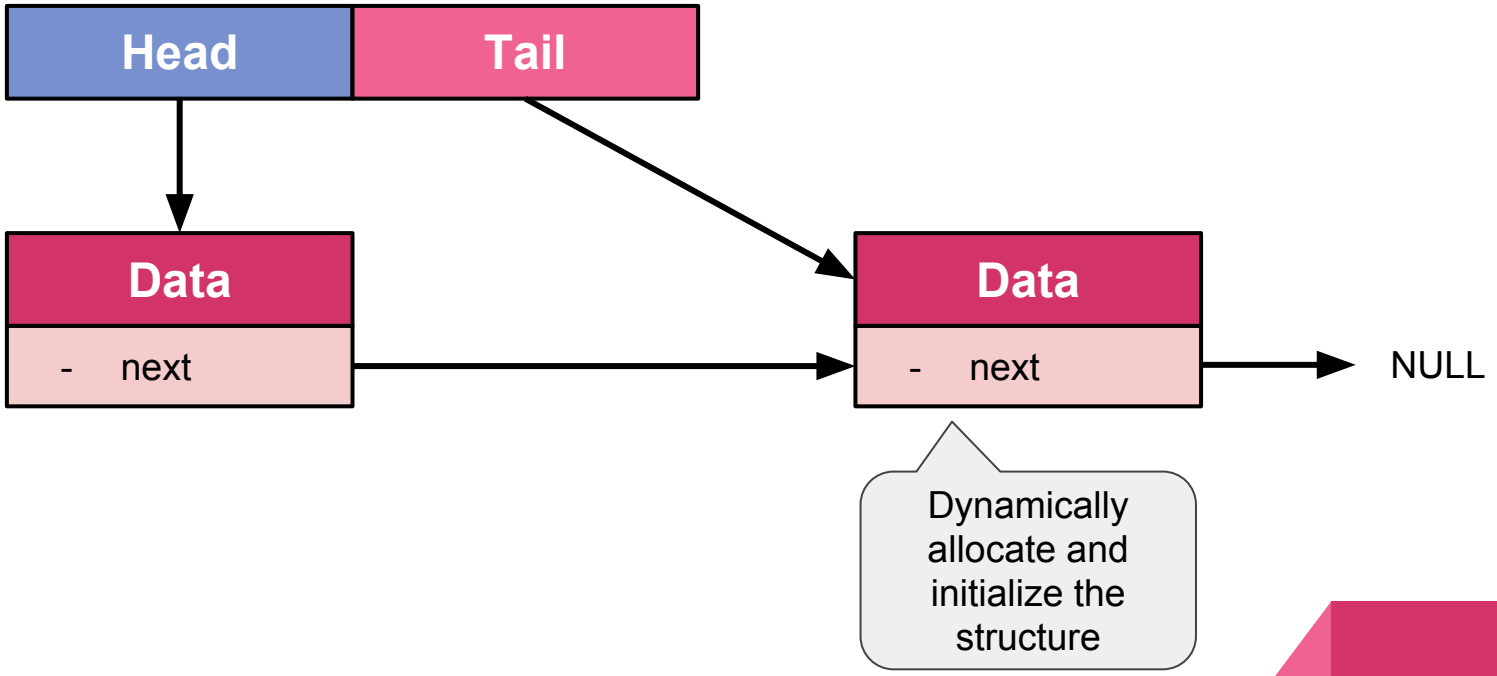
Problem 2: Suggestions

Adding an entry to the list!

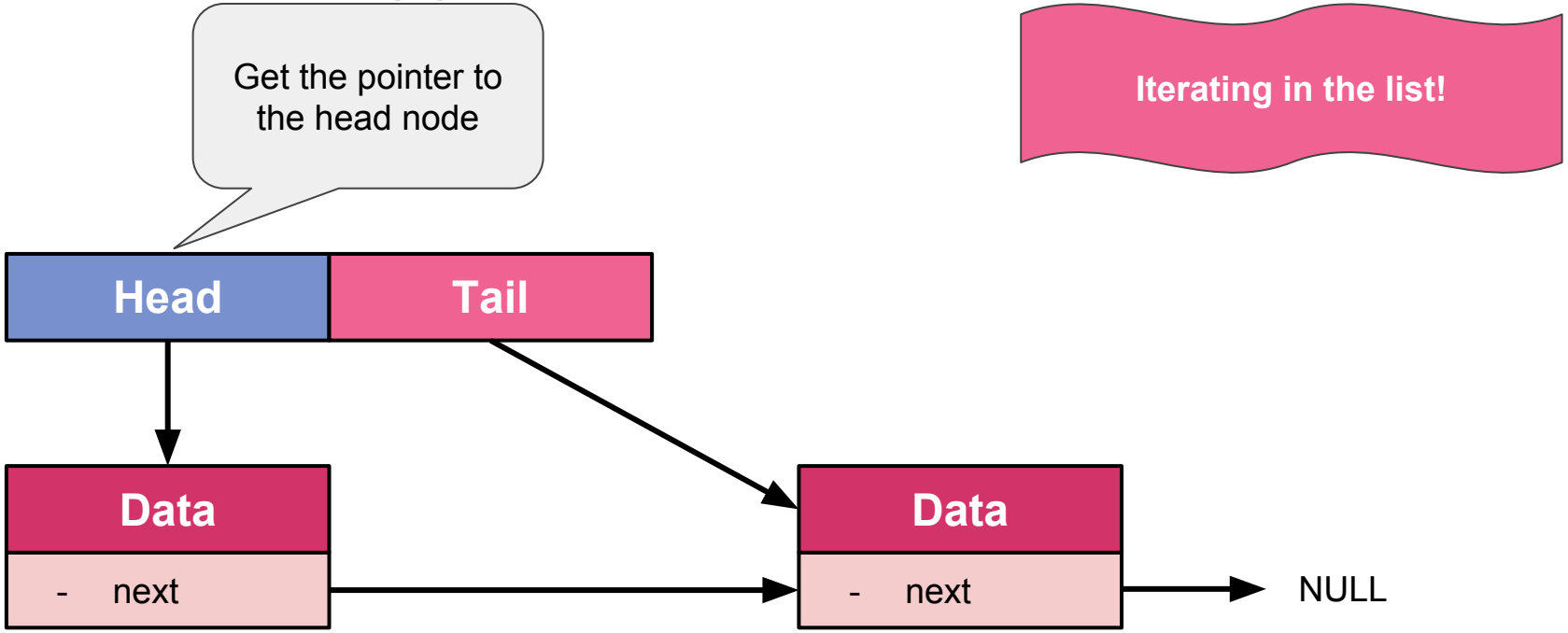


Problem 2: Suggestions

Adding an entry to the list!

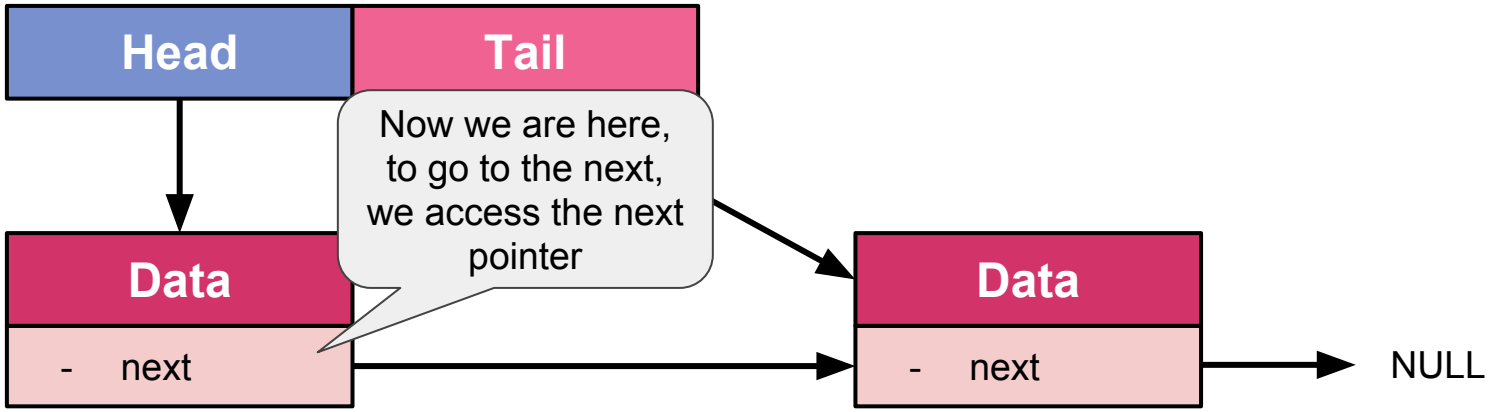


Problem 2: Suggestions



Problem 2: Suggestions

Iterating in the list!



Problem 2: Suggestions