C Programming Language

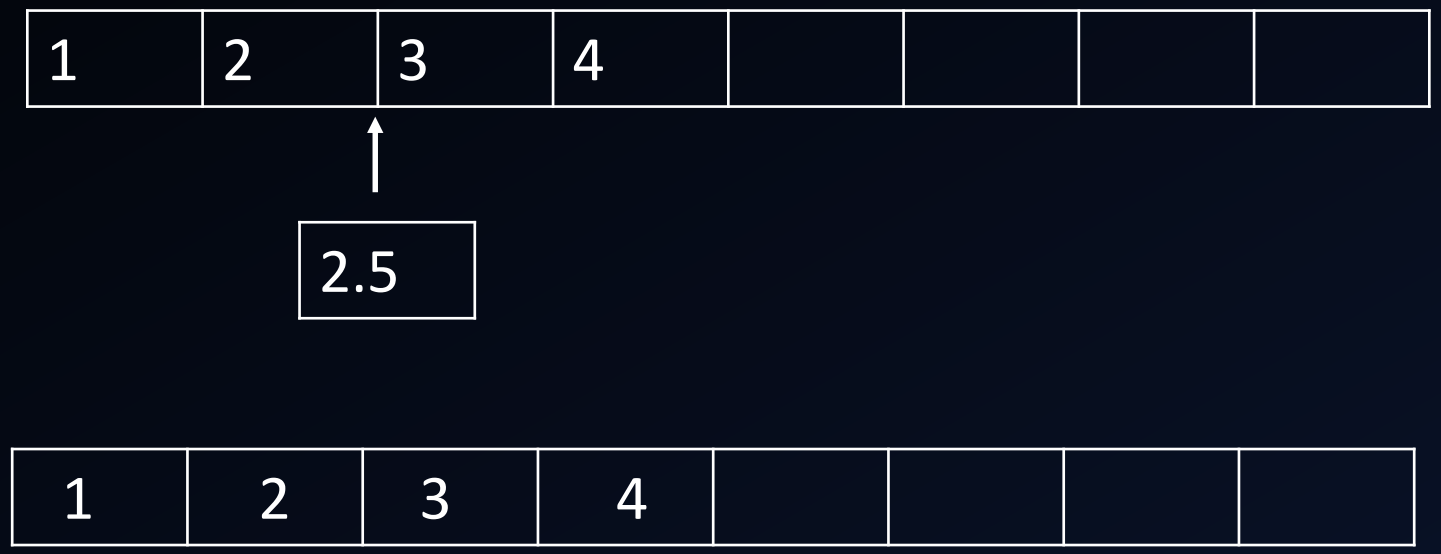
JANUARY 2, 2015

Today’s task

- Link list
- Recursion

Question

- How to insert an element in an array?



Link list

Array

Continuous memory

Link list

non Continuous memory

A value
A pointer

Link list

Define

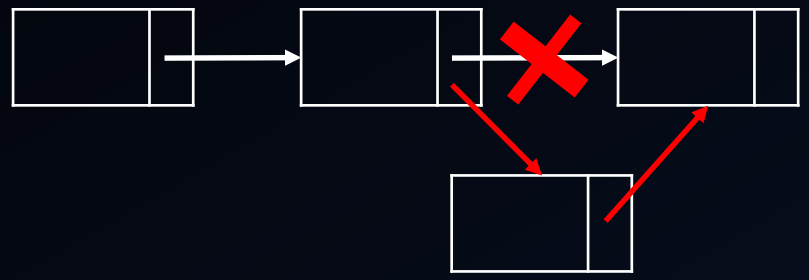
```
typedef struct Node {
    int x;
    struct Node *next;
}Node;
```

Traverse

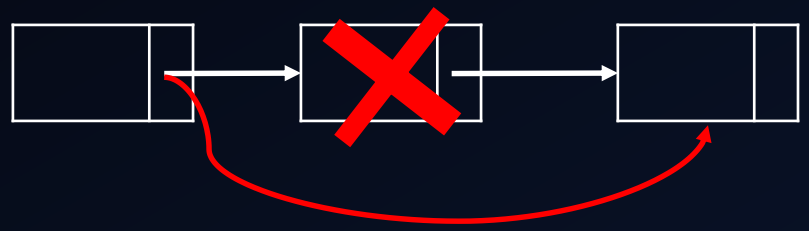
```
Node *conductor = root;
while(conductor)
{
        printf("%d\n",head->x);
        conductor = conductor->next;
}
```

Link list

- Insert



- Remove



Recursion

- How to compute n! ?
- n! = n * (n-1) * (n-2) *…* 2 * 1
- n! = n * (n-1)!
- (n-1)! = (n-1) * (n-2)!

- Recursion is a function that will call itself to finish the job

Recursion

- Use recursion to sum all values in a linked list

Recursion or Loop

- Both work
- Recursion usually makes code clean and simple

```
double fraction(int n)
{
        if(n==0)
            return 1;
        return n*fraction(n-1);
}
```

```
double fraction(int n)
{
        if(n==0)
            return 1;
        double out = 1;
        for (int i = 1; i <= n; ++i)
        {
            out *= i;
        }
        return out;
}
```

However, recursion is not always better

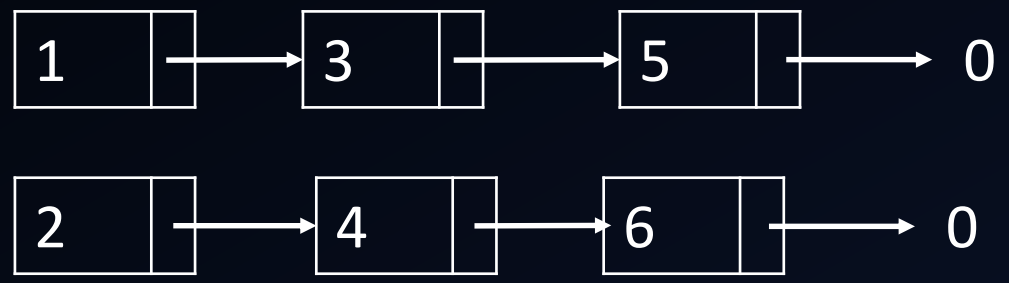
- E.g. To calculate Fibonacci Sequence
- Fibonacci Sequence
  - 0 1 1 2 3 5 8 13……
  - F(n) = F(n-1) + F(n-2)

```
int fibonacci(int n)
{
    if(n==0)
            return 0;
    if(n==1)
            return 1;
    return fibonacci(n-1) + fibonacci(n-2);
}
```

```
int fibonacci(int n)
{
        int n0 = 0;
        int n1 = 1;
        if(n==0)
        {
                return 0;
        }
        int i = 1;
        int out=0;
        while(i<n)
        {
                out = n0 + n1;
                n0 = n1;
                n1 = out;
                i++;
        }
        return out;
}
```

Homework

- Suppose we have two sorted linked list, merge the two list together and keep the new linked list still sorted
- E.g.



- Merged result

Next time

- Binary tree
- va_list