

Recursion (II)

Wagner Truppel
Lecturer, Dept. of Computer
Science & Engineering
UC Riverside

wagner@cs.ucr.edu
<http://www.cs.ucr.edu/~wagner>

<http://www.cs.ucr.edu/cs12>

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 1

Today's Topics

- Important example: binary search
- Fun example: towers of Hanoi
- Tail recursion
- Converting a recursive function to an iterative one

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 2

Binary Search

```
Book bookA[16]; // in this example, we have only 16 books
// suppose the array has been initialized already
Book targetBook = ...; // some book we want (say, book 62)

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
```

15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 3

Binary Search

hidx → 15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
lidx → 0	34

Book bookA[16]; // in this example, we have only 16 books
 // suppose the array has been initialized already
 Book targetBook = -; // some book we want (say, book 62)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
            
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 4

Binary Search

hidx → 15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
lidx → 0	34

Book bookA[16]; // in this example, we have only 16 books
 // suppose the array has been initialized already
 Book targetBook = -; // some book we want (say, book 62)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
            
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 5

Binary Search

15	89
14	83
13	78
12	78
hidx → 11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
lidx → 0	34

Book bookA[16]; // in this example, we have only 16 books
 // suppose the array has been initialized already
 Book targetBook = -; // some book we want (say, book 62)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
            
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 6

Binary Search

Book bookA[16]; // in this example, we have only 16 books
 // suppose the array has been initialized already
 Book targetBook = -; // some book we want (say, book 62)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 7

Binary Search

Book targetBook = -; // some book we want (say, book 89)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 8

Binary Search

Book targetBook = -; // some book we want (say, book 89)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

15	89
14	83
13	78
12	78
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 9

Binary Search

Book targetBook = _; // some book we want (say, book 89)

hidx → 15	89
14	83
13	78
12	76
midIdx → 11	71
10	64
9	63
8	62
lolIdx → 7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 10

Binary Search

Book targetBook = _; // some book we want (say, book 89)

hidx → 15	89
14	83
midIdx → 13	78
12	76
lolIdx → 11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 11

Binary Search

Book targetBook = _; // some book we want (say, book 89)

hidx → 15	89
midIdx → 14	83
lolIdx → 13	78
12	76
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 12

Binary Search

Book targetBook = _; // some book we want (say, book 89)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 13

Binary Search

Book targetBook = _; // some book we want (say, book 89)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

We're now stuck with an infinite loop!

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 14

Binary Search

Book targetBook = _; // some book we want (say, book 89)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 15

Binary Search

Book targetBook = _; // some book we want (say, book 87)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 16

Binary Search

Book targetBook = _; // some book we want (say, book 87)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 17

Binary Search

Book targetBook = _; // some book we want (say, book 87)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 18

Binary Search

Book targetBook = _; // some book we want (say, book 87)

15	89
14	83
midIdx → 13	78
12	76
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 19

Binary Search

Book targetBook = _; // some book we want (say, book 87)

15	89
14	83
midIdx → 14	83
13	78
12	76
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 20

Binary Search

Book targetBook = _; // some book we want (say, book 87)

15	89
14	83
midIdx → 14	83
13	78
12	76
11	71
10	64
9	63
8	62
7	61
6	49
5	48
4	47
3	42
2	39
1	35
0	34

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 21

Binary Search

Book targetBook = ...; // some book we want (say, book 87)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
    else
        return -1; // targetBook not found
}
    
```

We're stuck again with an infinite loop!

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 22

Binary Search

Book targetBook = ...; // some book we want (say, book 87)

```

// returns where in the array our book is (that is,
// returns the index of targetBook in the array), or
// -1 if the book isn't present in the array
int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (midIdx == loIdx)
        return -1; // targetBook not found
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 23

Towers of Hanoi

- Demo:
 - ◆ Java applet and stand-alone application
 - ◆ If trying the applet, need to download a plugin
 - ◆ Better simply to download jar file
 - ◆ Need Java 1.2 or better installed

```

void hanoi(int nDisks, Pole src, Pole dest, Pole aux)
{
    if (nDisks < 1)
        print error message;
    else
        if (nDisks == 1)
            move top disk from source to destination;
        else // nDisks > 1
            {
                hanoi(nDisks - 1, src, aux, dest);
                move top disk from source to destination;
                hanoi(nDisks - 1, aux, dest, src);
            }
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 24

Tail recursion

- **Recursion** incurs on some overhead b/c of time and memory required to construct the activation frame for each recursive call
- **Iteration** has a much lower overhead and is faster
- Why use recursion then ?
 - ◆ Sometimes it's easier to come up with a recursive solution to a problem than an iterative one
 - ◆ Sometimes the only *practical* solution is recursive
- Can we convert recursive algorithms into iterative ones ?
- Yes !

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 25

Tail recursion

- Can we convert recursive algorithms into iterative ones ? Yes !
- It's easy when the recursive function is executed only once in its own body
- It's even easier when that one call is the **last** executed statement (**Tail Recursion**)

```

int BinS(int loIdx, int hiIdx)
{
    int midIdx = (loIdx + hiIdx) / 2;

    if (targetBook == bookA[midIdx])
        return midIdx; // found it!
    else if (targetBook == bookA[hiIdx])
        return hiIdx; // found it!
    else if (midIdx == loIdx)
        return -1; // targetBook not found
    else if (targetBook < bookA[midIdx])
        return BinS(loIdx, midIdx); // search lower half
    else if (targetBook > bookA[midIdx])
        return BinS(midIdx, hiIdx); // search upper half
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 26

Tail recursion

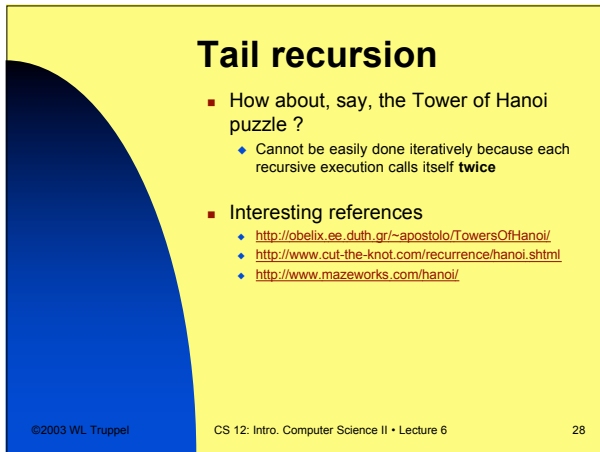
- Iterative version of binary search

```

int BinS(int loIdx, int hiIdx)
{
    while (true)
    {
        int midIdx = (loIdx + hiIdx) / 2;

        if (targetBook == bookA[midIdx])
            return midIdx; // found it!
        else if (targetBook == bookA[hiIdx])
            return hiIdx; // found it!
        else if (midIdx == loIdx)
            return -1; // targetBook not found
        else if (targetBook < bookA[midIdx])
            hiIdx = midIdx; // search lower
        else if (targetBook > bookA[midIdx])
            loIdx = midIdx; // search upper
    }
}
    
```

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 27



Tail recursion

- How about, say, the Tower of Hanoi puzzle ?
 - ◆ Cannot be easily done iteratively because each recursive execution calls itself **twice**
- Interesting references
 - ◆ <http://obelix.ee.duth.gr/~apostolo/TowersOfHanoi/>
 - ◆ <http://www.cut-the-knot.com/recurrence/hanoi.shtml>
 - ◆ <http://www.mazeworks.com/hanoi/>

©2003 WL Truppel CS 12: Intro. Computer Science II • Lecture 6 28
