Lesson 6 The power of halving and doubling

# Learning objectives

1. Derive the run-time of algorithms that iteratively cut the size of the input in half.
2. Realize that the strategy of halving or doubling can often produce more efficient algorithms than the obvious straight-forward approach.

# Run-time analysis of binary search

Recall the binary search algorithm:

**binarySearch**(l, h, x, a) // input size = h – l

if l = h

Stuff1

else

Stuff2

if Stuff3

**binarySearch**(mid, h, x, a) //input size = h – mid = (h – l)/2

else

**binarySearch**(l, mid, x, a) //input size = mid – l = (h – l)/2

Let T(n) = the worst-case time it takes for binarySearch to find an item in a list of n items

Let s = the time to do Stuff2 and Stuff3 (it’s constant)

Assume T(1) = c

T(n) = T(n/2) + s

= T(n/4) + s + s

= T(n/4) + 2s Can you take it from here?

= T(n/8) + s + 2s

= T(n/8) + 3s

= …

= T(1) + ks 🡪 1 = n/2k means that k = log2 n

= c + ks

= c + s log2n 🡪 log n is the dominant term, and s is a constant.

= O( log n )

Therefore, binary search algorithm runs in O(log n) time.

# The Master Theorem (simplified version)

Suppose an algorithm cuts the input size in half in each iteration.

If the number of steps to achieve this cut is:

* Linear in n, then T(n) = O(n log n) 🡪 Example: merge sort
* Constant, then T(n) = O(log n) 🡪 Example: binary search

Look up the adult-version of The Master Theorem online.

# Moral:

When designing algorithms, consider trying a divide-and-conquer approach.

# The Hiker and the Bridge problem

You’re hiking along the river in a dark forest at night (as opposed to a bright forest at night), and you’re looking for a bridge that will get you across the river.

You don’t know if the bridge is ahead of you or behind you.

You have nothing with you except a small flashlight.

The river is too cold and too swift to attempt to swim.

Goal: Design an algorithm for finding the bridge with a minimum amount of walking.

## Take up student solutions. Then present the doubling-solution