Recall Dynamic Resizing's Approach to Array Sizes

Think back to dynamic resizing:
- we double our array
- each time we need more.

When we examined waste space,
- we found that doing so
- gave us a pretty good fit
- (average 38% waste).

We Build a Best-Fit Logarithmic Allocator

Let's use the same idea for allocation:
- **allocate the smallest power of 2 bytes**
- **into which** the desired **block fits**.

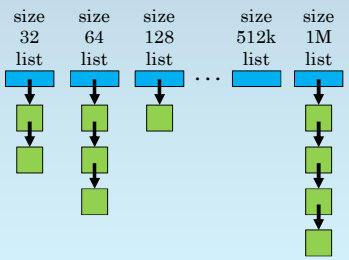
This approach is called a **best-fit logarithmic allocator**.

We **might allow blocks to be split** (into two smaller blocks) **and re-combined**.
- For example, see the page allocation management in the Linux kernel (in ECE391).
- **Our implementation does neither.**

A Linked List Holds Free Blocks of Each Size

Let's talk about data structures.

Free blocks are kept in **linked lists based on the size of the blocks**, as shown to the right.



Allocate New Blocks as Necessary (As Done Earlier)

When we need a block, we **look in the list**.

For example, if we want 100 bytes, we look in the size 128 list.

If list is empty, we **allocate a new block** (as before).