More practice with recursion

Efficient GCD algorithm (greatest common divisor)

Observation: let $a, b \in \mathbb{Z}$. Then

$\exists$ $q, r \in \mathbb{Z}$, $r \lneq b$ s.t.

"there exists"

$$\underline{a = qb + r.}$$ (Division Algorithm)

Observation: common divisors of a, b are the same as those of b, r. $\otimes$

Proof: Suppose $d \mid a$ and $d \mid b$.

"
($\exists k \in \mathbb{Z}$ s.t. $a = dk$.)

Then $d \mid r$:

$d \mid a \Rightarrow \exists k \in \mathbb{Z}$ s.t. $a = dk$

$d \mid b \Rightarrow \exists k' \in \mathbb{Z}$ s.t. $b = dk'$

Since $\underline{a = qb + r}$,
$$\begin{aligned} r &= a - qb \\ &= dk - qdk' \\ &= d(k - qk') \end{aligned}$$

So $d \mid r$, since $k - qk' \in \mathbb{Z}$. ✓

And if $d \mid b$ & $d \mid r$, of course $d \mid a$:

$a = qb + r$

so if $r = dk$
$b = dk'$, then $a = d(\underbrace{qk' + k}_{\mathbb{Z}})$. ✓

We can use observation $\otimes$ as the basis of a recursive algo for gcd's:

$$\gcd(a,b) = \gcd(b,r),$$

and $\underline{r \underset{\neq}{<} b}$.

Recall the meta:

1. Solve small instances explicitly ("base case")
2. Pretend your function works on all <u>smaller</u> inputs, and use that to build the answer to your instance.

Question what do we consider the "size" of a gcd instance a,b?
We'll use $|b|$ (size of <u>second</u> input).
Base case? $\underline{b = 0}$.

```
int gcd(int a, int b)
{
    // base case: gcd(a,0) = a
```

```
    if (b == 0) return a;
    return gcd(b, a%b);
}
```

Example: $\overset{a}{(12}, \overset{b}{18)} = (\overset{b}{18}, \overset{a\%b}{12})$

$(\overset{a}{18}, \overset{b}{12}) = (\overset{b}{12}, \overset{a\%b}{6})$

$(\overset{a}{12}, \overset{b}{6}) = (\overset{b}{6}, \overset{a\%b}{0})$

$(\overset{a}{6}, \overset{b}{0}) = 6$ ✓

Ex. 2: $(7, 11) = (11, 7)$

$(11, 7) = (7, 4)$

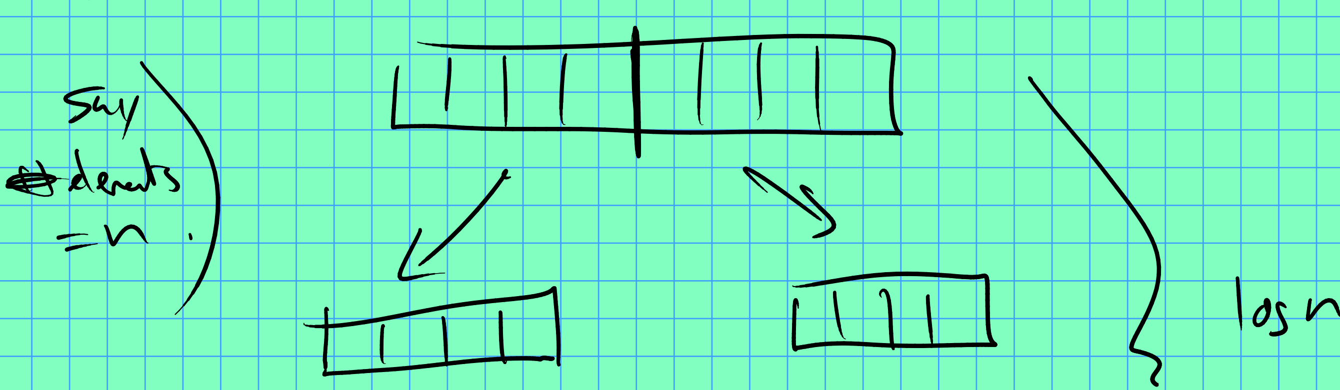
$(7, 4) = (4, 3)$

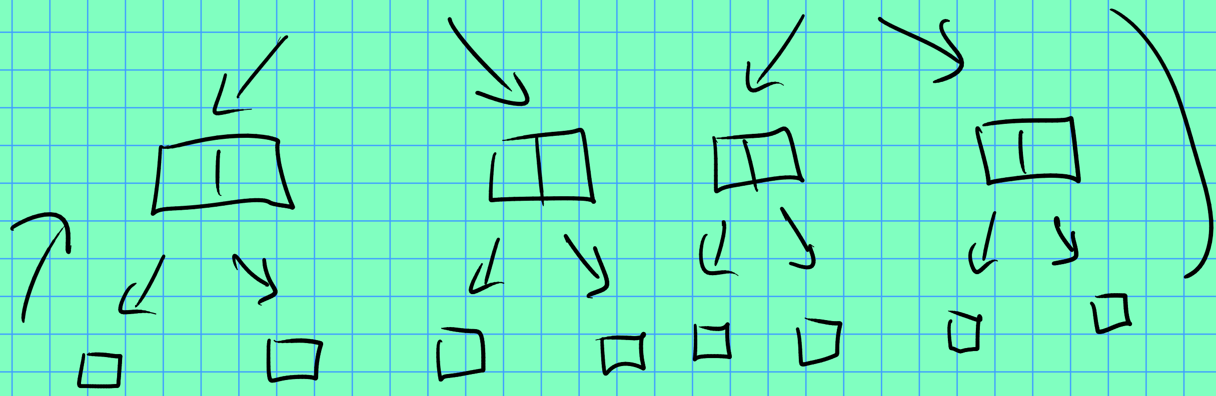
$(4, 3) = (3, 1)$

$(3, 1) = (1, 0)$

$(1, 0) = 1$ ✓

Back to merge sort:





$4 \cdot 2 = 8 = n$

$2 \cdot 4 = 8 = n$

$1 \cdot 8 = 8 = n$

Total cost $\approx n \cdot \log_2 n$.