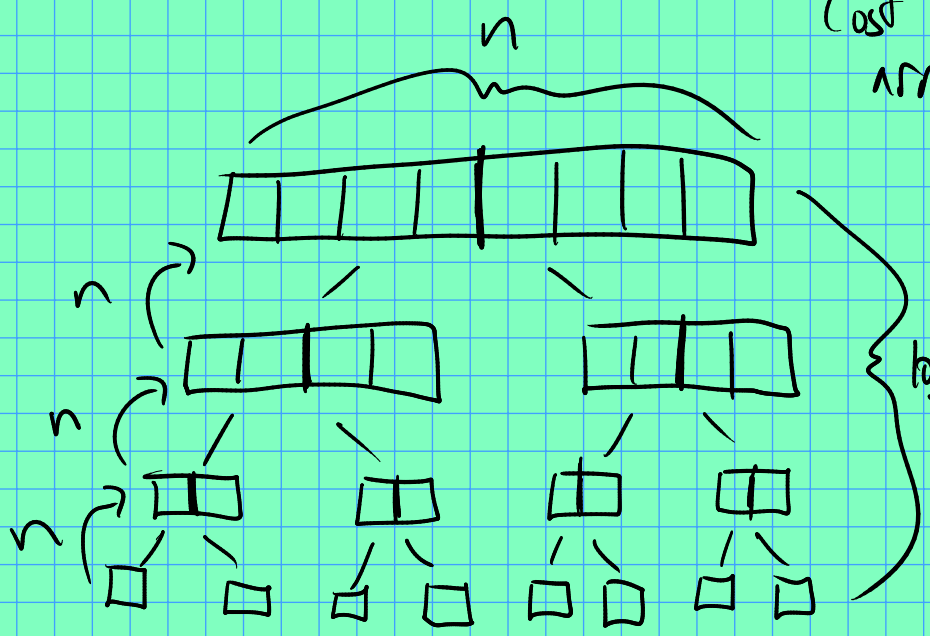

Cost to merge 2 $n/2$ sized arrays? $\approx n$

$\left(\left(\frac{n}{2}\right)/2\right)/2 \cdots = 1$

$\frac{n}{2^k} = 1$

$n = 2^k$

$\log_2 n = k$

$\therefore$ Total cost of Merge sort $\approx n \cdot \log_2 n$.

Note: this is way better than selection sort or insertion sort for large inputs!!

(Both took $\approx n^2 = n \cdot n \ll n \cdot \log_2 n$ (for large n))

New thing: **Maps** (aka "Associative Arrays" or "Dictionaries")

Could think of these as "vectors indexed by (possibly) non-integer types"

Or

partially defined function

Or

lookup table of values ...

Motivating example

Make a frequency table of strings given on stdin.

```
echo aaa bbb aaa ccc bbb aaa | ./freq
```

output:

```
aaa: 3
bbb: 2
ccc: 1
```

Note: maps allocate space as needed.

Also, they initialize integer values to 0 for you!

```
x = F["xyz"]  // if "xyz" never accessed before,
              // F["xyz"] will be created + set to 0.
```

(Aside: how many strings of length 10 are there?

$(2^8)^{10} = 2^{80}$)

Note: the indexes are unique. There can only be one F[x] per x.

Note: when declaring a map<T, T'> for data types T, T':

T' can be anything you want;

T must have a '<' operator defined.