

Recall OS/notes.pdf...

Want to compute

$$x_0 \boxed{?} x_1 \boxed{?} \dots \boxed{?} x_{n-1}$$

for some binary operation $\boxed{?}$.

⊗ What if we could store a function into a variable?

Sketch for general "fold" function:

```
int fold (vector<int> v, op  $\boxed{?}$ , int start)
{
    int s = start;
    for (i = 0; i < v.size(); i++)
        s =  $\boxed{?}$ (s, v[i]);
    return s;
}
```

Strings

String type is basically `vector<char>`,
but with nice extra functions:

— `cout << s` works!

— "concatenation" `s + t` (`+=` works too)

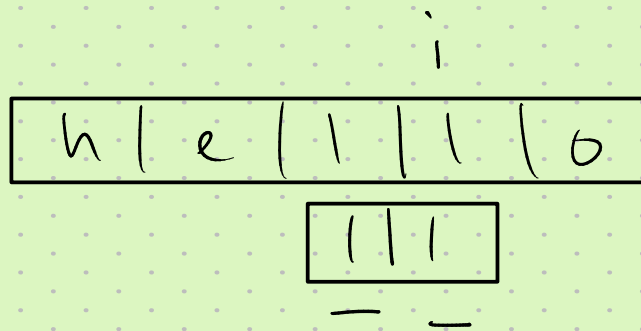
— "Ctrl-f", / : can search through
strings using `s.find(...)`

Note: `s.length()`
gives `s.size()`.

Let's write our own version of find:

```
size_t find(const string& s, const string& t);  
// Search for t as a substring in s.
```

Idea?



Idea: look for match at all possible starting indexes. (Think sliding window...)

Possible starting points for match in s?

$0, 1, \dots, |s| - |t|$ (inclusive)

```
for (i = 0; i <= |s| - |t|; i++) {  
    // check for match @ i  
    for (j = 0; j < |t|; j++) {  
        if (s[i+j] != t[j]) break;  
    }  
    ... next time ...
```