CORE C++
LIGHTENING STORM

INTERESTING ALGORITHMS (JUST SAY NO TO RAW LOOPS)

Shimon Shore

Chief Architect, Open Source Program

Israel Ministry of Science and Technology

Israel Ministry of Culture and Sport

BASED ON:
BETTER CODE (C++ SEASONING)

SEAN PARENT | PRINCIPAL SCIENTIST

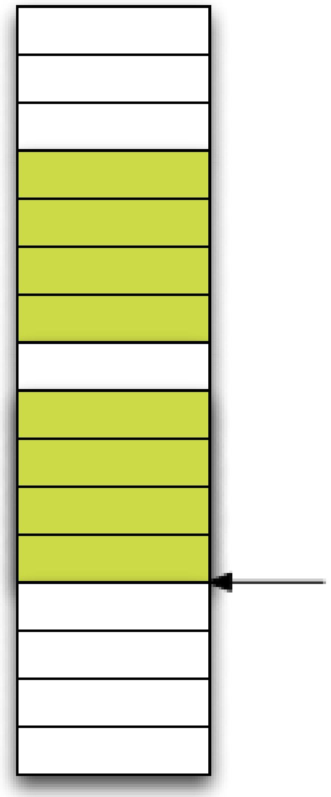
No Raw Loops

WHAT IS A RAW LOOP?

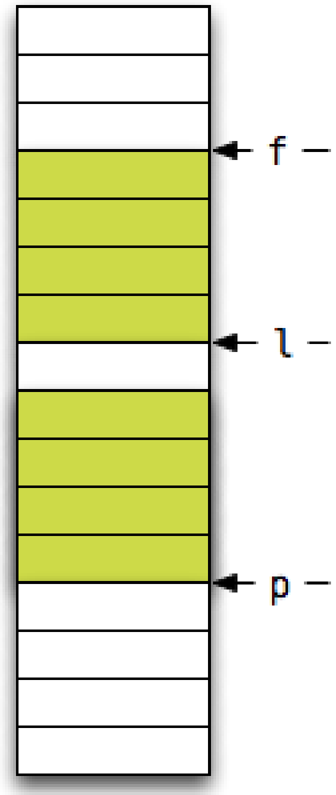
- A *raw loop* is any loop inside a function where the function serves purpose larger than the algorithm implemented by the loop

Two Beautiful Examples

NO RAW LOOPS

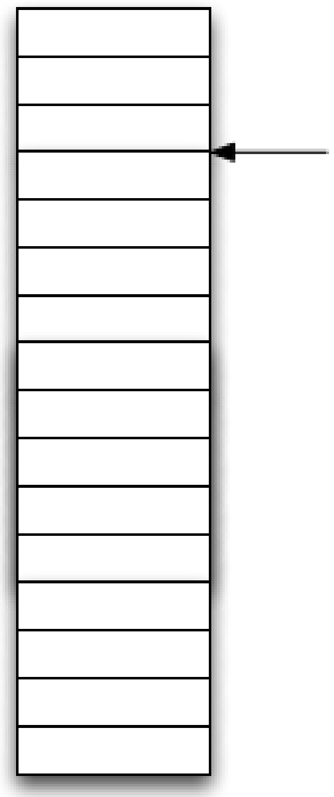


NO RAW LOOPS

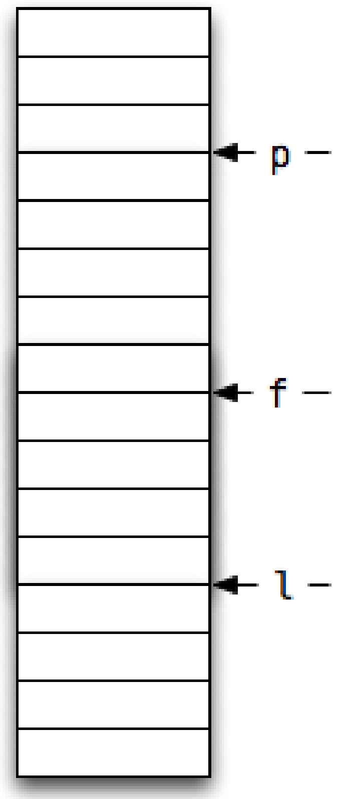


```
rotate(f, l, p);
```

NO RAW LOOPS

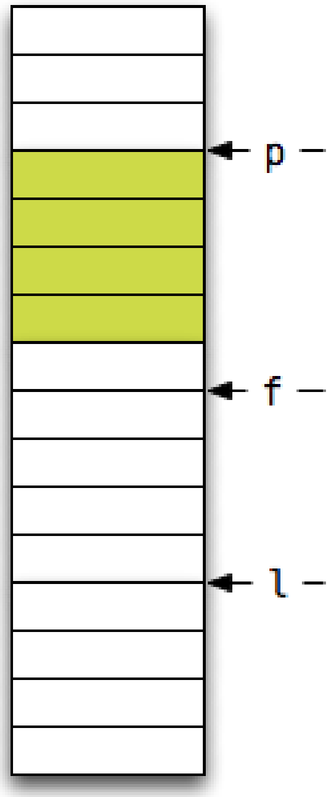


NO RAW LOOPS



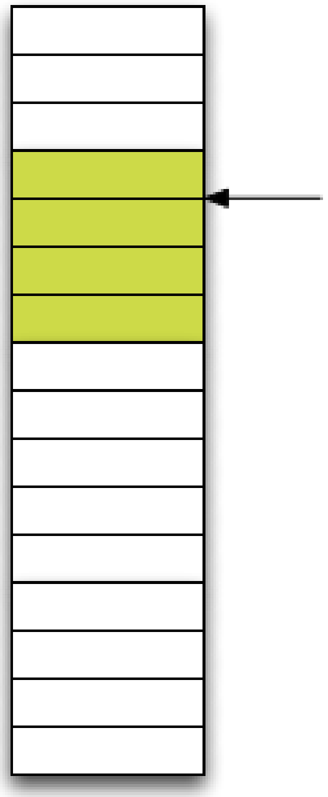
rotate(p, f, l);

NO RAW LOOPS



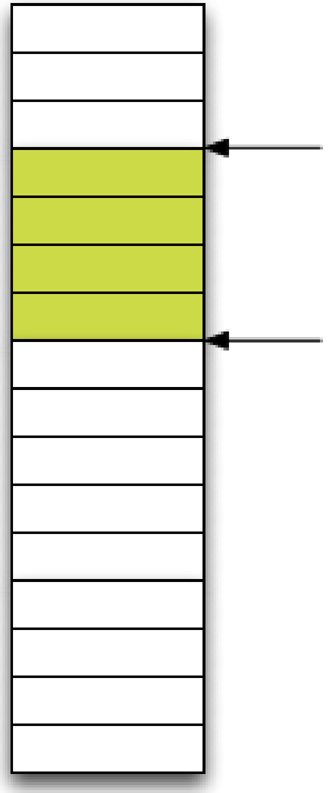
```
if (p < f) rotate(p, f, l);
if (l < p) rotate(f, l, p);
```

NO RAW LOOPS



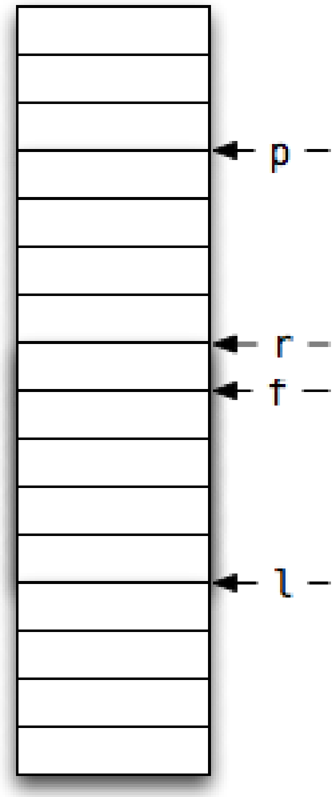
```
if (p < f) rotate(p, f, l);
if (l < p) rotate(f, l, p);
```

NO RAW LOOPS



```
if (p < f) rotate(p, f, l);
if (l < p) rotate(f, l, p);
```

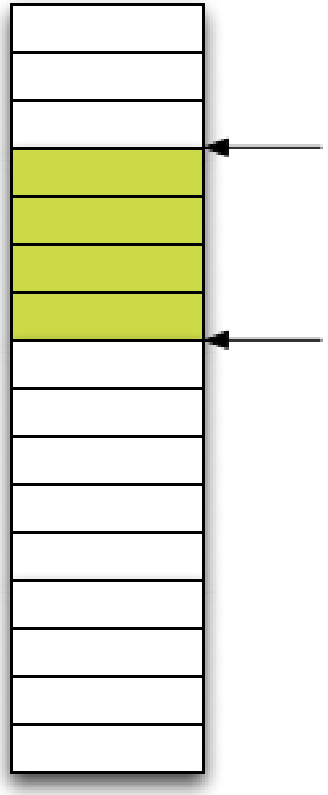
NO RAW LOOPS



```
if (p < f) return { p, rotate(p, f, l) };
if (l < p) return { rotate(f, l, p), p };
```

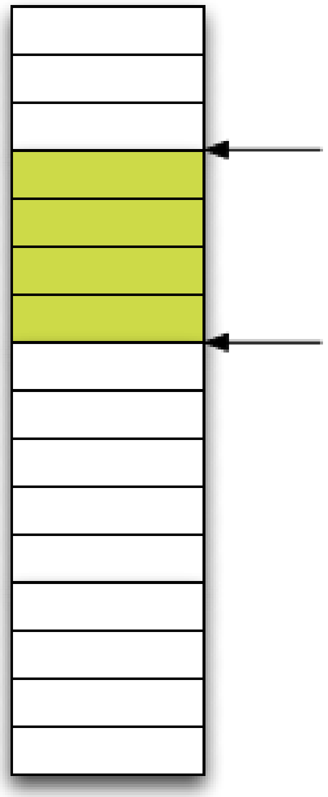
C++11

NO RAW LOOPS



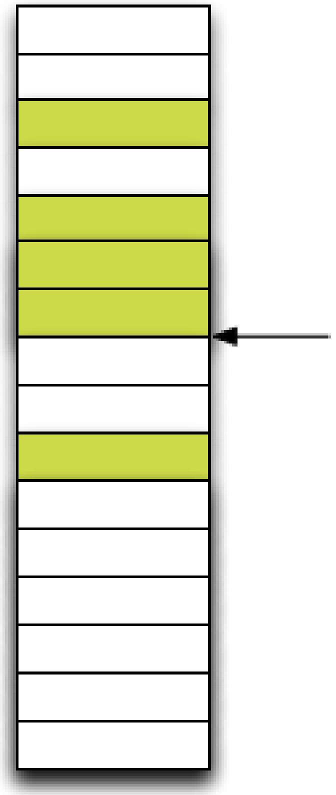
```
if (p < f) return { p, rotate(p, f, l) };
if (l < p) return { rotate(f, l, p), p };
return { f, l };
```

NO RAW LOOPS

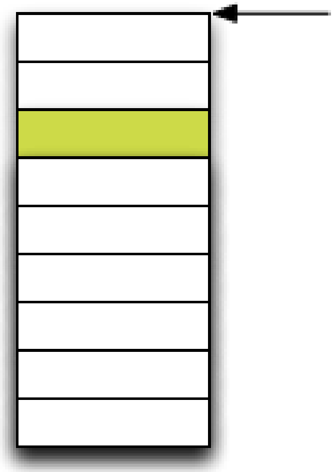


```
template <typename I> // I models RandomAccessIterator
auto slide(I f, I l, I p) -> pair<I, I>
{
   if (p < f) return { p, rotate(p, f, l) };
   if (l < p) return { rotate(f, l, p), p };
   return { f, l };
}
```

NO RAW LOOPS

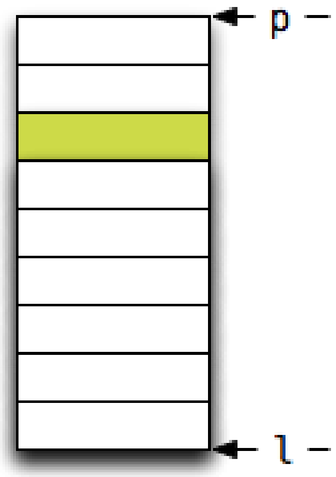


NO RAW LOOPS

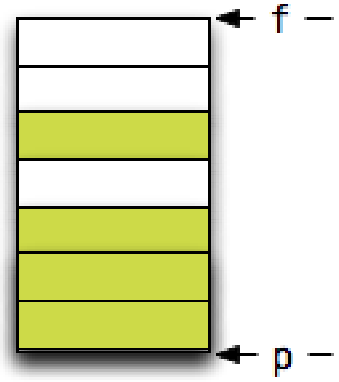


NO RAW LOOPS

stable_partition(p, l, s)

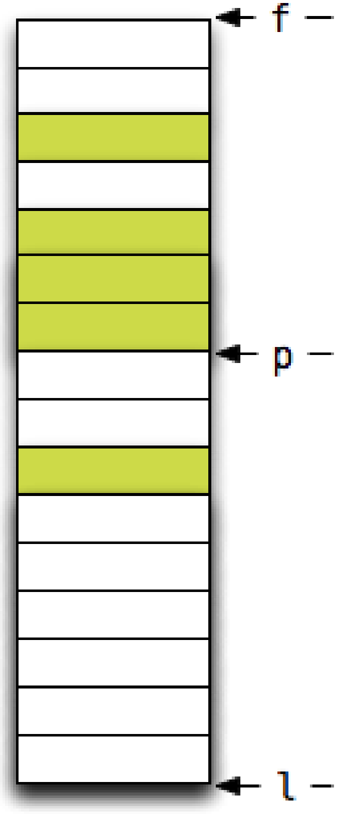


NO RAW LOOPS



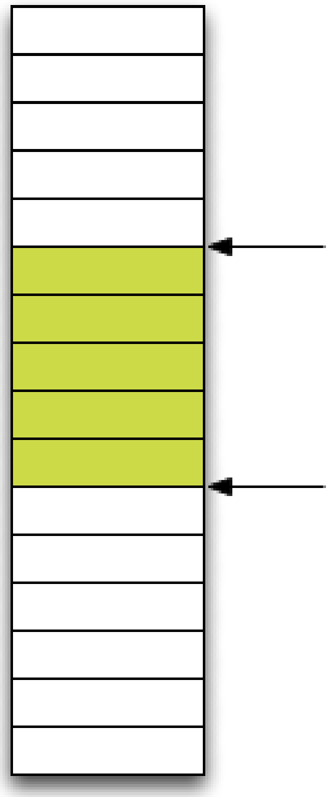
```
stable_partition(f, p, not1(s))
```

NO RAW LOOPS



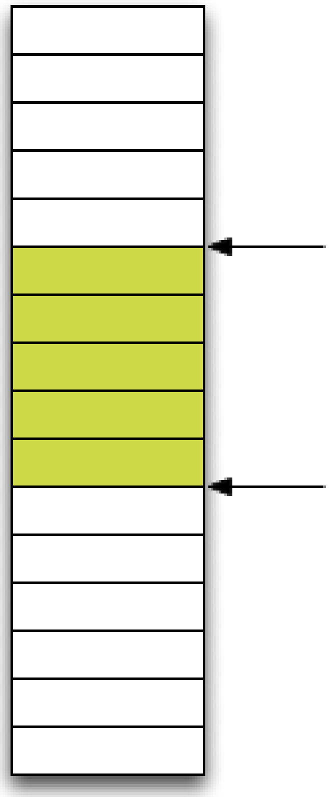
```
stable_partition(f, p, not1(s))
stable_partition(p, l, s)
```

NO RAW LOOPS



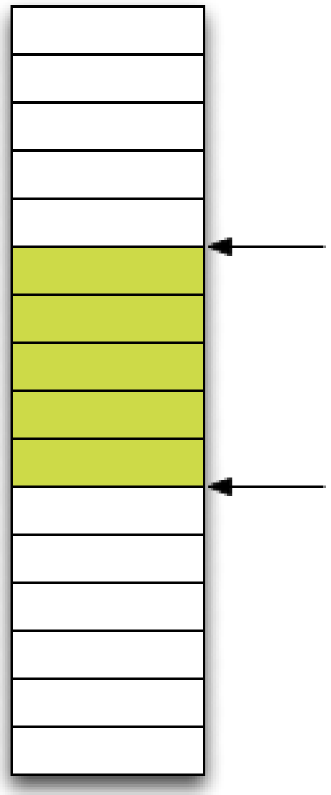
```
stable_partition(f, p, not1(s))
stable_partition(p, l, s)
```

NO RAW LOOPS



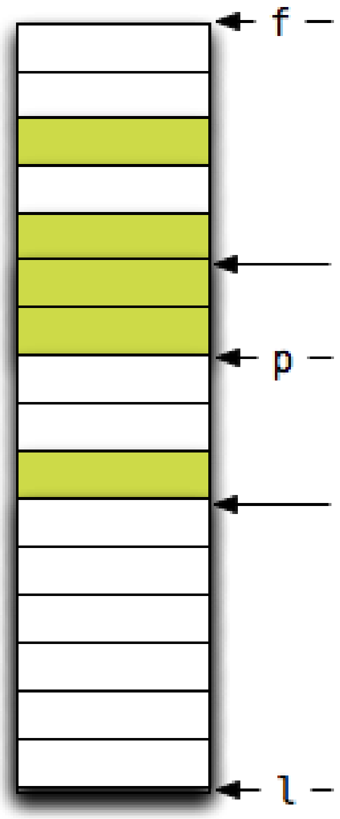
```
return { stable_partition(f, p, not1(s)),
         stable_partition(p, l, s) };
```

NO RAW LOOPS



```
template <typename I, // I models BidirectionalIterator
        typename S> // S models UnaryPredicate
auto gather(I f, I l, I p, S s) -> pair<I, I>
{
   return { stable_partition(f, p, not1(s)),
            stable_partition(p, l, s) };
}
```

NO RAW LOOPS



```
template <typename I, // I models BidirectionalIterator
          typename S> // S models UnaryPredicate
auto gather(I f, I l, I p, S s) -> pair<I, I>
{
   return { stable_partition(f, p, not1(s)),
            stable_partition(p, l, s) };
}
```

NO RAW LOOPS

```
template <typename I, // I models BidirectionalIterator
          typename S> // S models UnaryPredicate
auto gather(I f, I l, I p, S s) -> pair<I, I>
{
    return { stable_partition(f, p, not1(s)),
             stable_partition(p, l, s) };
}
```

*

LIVE

CORE C++

INTERESTING ALGORITHMS

June 28th

SHIMON SHORE | JUST SAY NO TO RAW LOOPS