

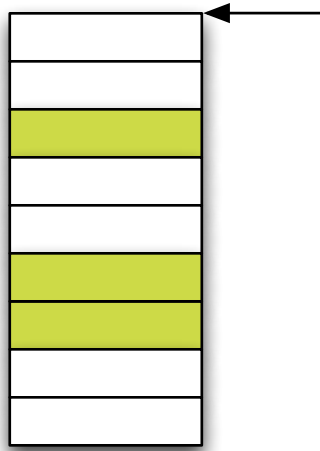
No Row Loops



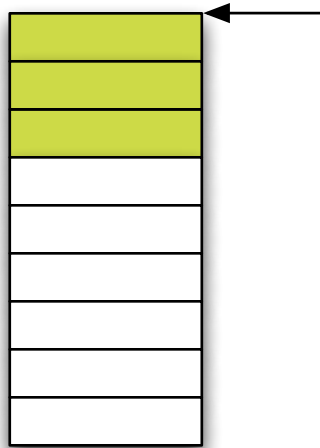
No Row Loops



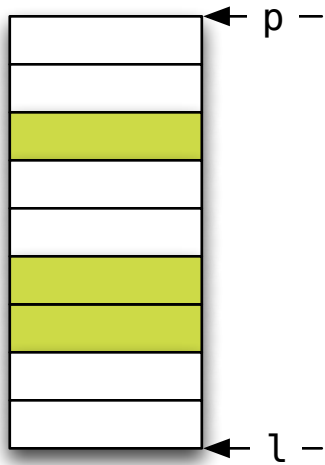
No Row Loops



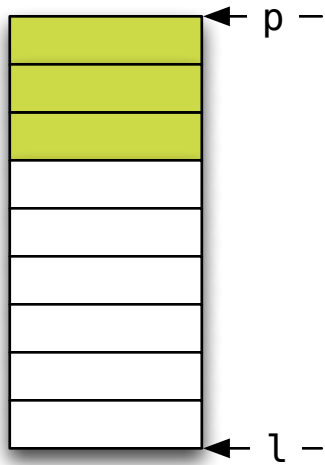
No Row Loops



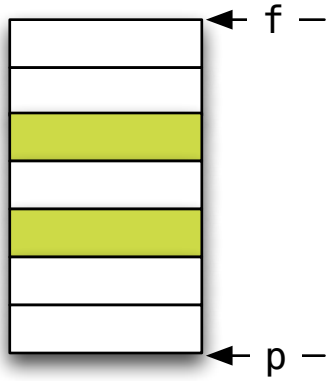
```
stable_partition(p, l, s)
```



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

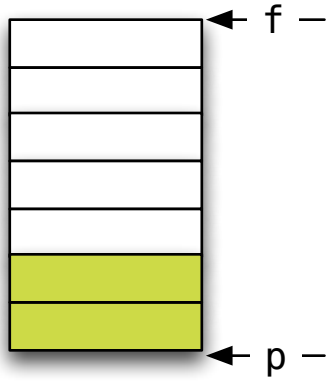


No Row Loops



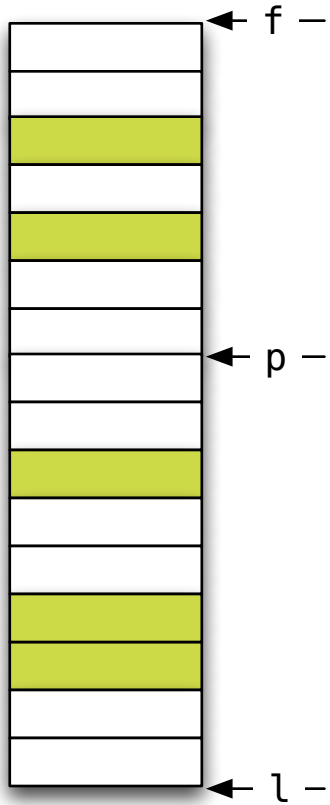
```
stable_partition(f, p, not1(s))
```

No Raw Loops



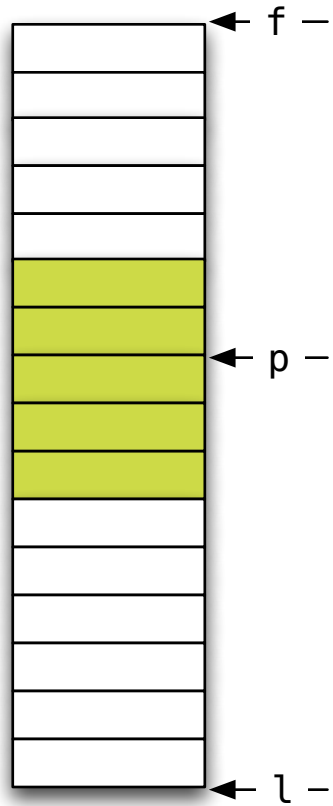
```
stable_partition(f, p, not1(s))
```


No Row 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 Row Loops



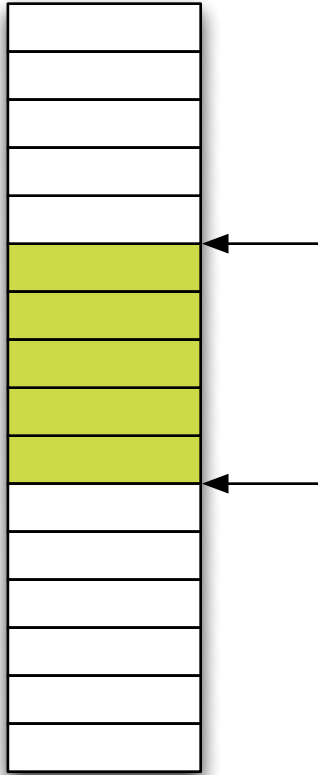
```
stable_partition(f, p, not1(s))  
stable_partition(p, l, s)
```

No Row Loops



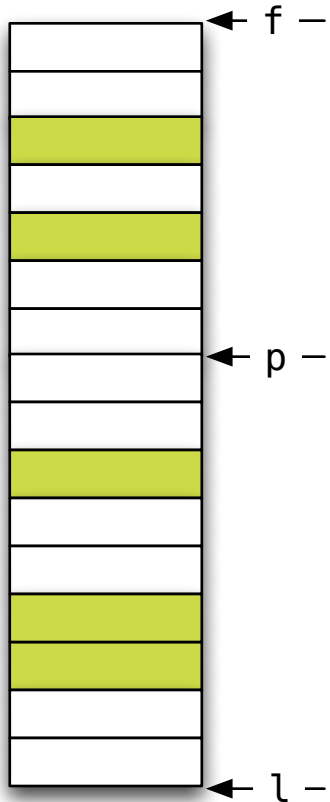
```
return { stable_partition(f, p, not1(s)),  
         stable_partition(p, l, s) };
```

No Raw Loops



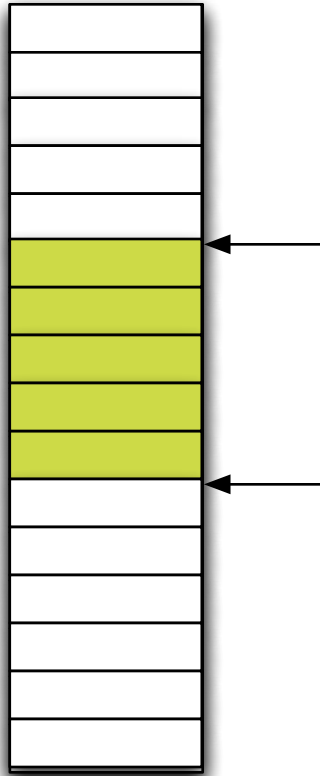
```
template <typename I, // I models BidirectionalIterator
          typename S> // S models UnaryPredicate
auto gather(I f, I l, I p, S s) -> pair<I, I>
{
    return { stable_partition(f, p, not1(s)),
             stable_partition(p, l, s) };
}
```

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) };
}
```