Optimality of FF

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions. i.e., pages only brought in upon request (cache miss)

<table>
<thead>
<tr>
<th>Sequence of requests</th>
<th>Cache changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>e a f</td>
<td></td>
</tr>
<tr>
<td>e a g</td>
<td>g brought in</td>
</tr>
<tr>
<td>g requested</td>
<td></td>
</tr>
<tr>
<td>b a g</td>
<td>b brought in</td>
</tr>
<tr>
<td>b a g</td>
<td></td>
</tr>
<tr>
<td>b a g</td>
<td></td>
</tr>
<tr>
<td>b a g</td>
<td></td>
</tr>
<tr>
<td>b a g</td>
<td></td>
</tr>
</tbody>
</table>
**Optimality of FF**

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions.

<table>
<thead>
<tr>
<th>Sequence of requests</th>
<th>Cache changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>e a f</td>
<td>g brought in</td>
</tr>
<tr>
<td>e a f</td>
<td></td>
</tr>
<tr>
<td>g requested</td>
<td>e a g</td>
</tr>
<tr>
<td>g brought in</td>
<td></td>
</tr>
<tr>
<td>b requested</td>
<td>b a g</td>
</tr>
<tr>
<td>b brought in</td>
<td></td>
</tr>
</tbody>
</table>
Optimality of FF

**Lemma**: every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

<table>
<thead>
<tr>
<th>OPT</th>
<th>FF</th>
<th>OPT’</th>
</tr>
</thead>
<tbody>
<tr>
<td>x - x - -</td>
<td>x - x - -</td>
<td>x - x - -</td>
</tr>
</tbody>
</table>

choice
\[ k+1 \]
Optimality of FF

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact:** if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

<table>
<thead>
<tr>
<th>OPT</th>
<th>x - x - -</th>
<th>\text{evict e}</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>x - x - -</td>
<td>\text{evict f}</td>
</tr>
<tr>
<td>OPT’</td>
<td>x - x - -</td>
<td>choice $k+1$</td>
</tr>
</tbody>
</table>
Optimality of FF

**Lemma**: every eviction schedule can be "reduced" without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT' that is consistent with FF on the first $k+1$ choices.

```
OPT     x - x - -  xevict e
FF      x - x - -  xevict f
OPT'    x - x - -  xevict f
```

choice $k+1$
**Optimality of FF**

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact:** if reduced OPT & FF are consistent on first \( k \) choices, then there exists a reduced OPT’ that is consistent with FF on the first \( k+1 \) choices.

![Diagram showing eviction schedules](image)
**Optimality of FF**

**Lemma**: every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution:**

**Case 1**: e never requested again. Then f is never requested again, so resolution is trivial.
**Optimality of FF**

**Lemma**: every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution:**

**Case 1**: e never requested again. Then f is never requested again, so resolution is trivial.

**Case 2**: e’s next request is before f’s. Then OPT’ does what OPT does, except when OPT does anything to pages e or f.
**Optimality of FF**

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact:** if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution:**

**Case 1:** e never requested again. Then f is never requested again, so resolution is trivial

**Case 2:** e’s next request is before f’s. Then OPT’ does what OPT does, except when OPT does anything to pages e or f.

**Case 2.1:** OPT evicts f. Then OPT’ should evict e.
Optimality of FF

**Lemma**: every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution**:

**Case 1**: e never requested again. Then f is never requested again, so resolution is trivial.

**Case 2**: e’s next request is before f’s. Then OPT’ does what OPT does, except when OPT does anything to pages e or f.

- **Case 2.1**: OPT evicts f. Then OPT’ should evict e.

- **Case 2.2**: OPT evicts g to bring in e. If g=f,
Optimality of FF

**Lemma**: every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact**: if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution**:  

*Case 1*: $e$ never requested again. Then $f$ is never requested again, so resolution is trivial.

*Case 2*: $e$’s next request is before $f$’s. Then OPT’ does what OPT does, except when OPT does anything to pages $e$ or $f$.

- **Case 2.1**: OPT evicts $f$. Then OPT’ should evict $e$.
- **Case 2.2**: OPT evicts $g$ to bring in $e$.  
  - if $g=f$, OPT not optimal.  
  - if $g$ not $f$,  

See diagram for visual representation.
Optimality of FF

**Lemma:** every eviction schedule can be “reduced” without increasing the number of evictions.

**Main fact:** if reduced OPT & FF are consistent on first $k$ choices, then there exists a reduced OPT’ that is consistent with FF on the first $k+1$ choices.

**Cache resolution:**

**Case 1:** e never requested again. Then f is never requested again, so resolution is trivial.

**Case 2:** e’s next request is before f’s. Then OPT’ does what OPT does, except when OPT does anything to pages e or f.

**Case 2.1:** OPT evicts f. Then OPT’ should evict e.

**Case 2.2:** OPT evicts g to bring in e.

if $g=f$, OPT not optimal.
if $g$ not f, OPT’ evicts g to bring in f.

OPT’ not reduced!