

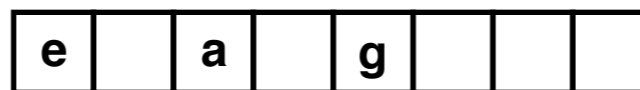
Optimality of FF

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

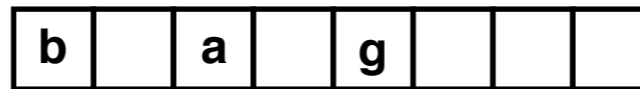
i.e., pages only brought in upon request (cache miss)

Sequence of requests

Cache changes

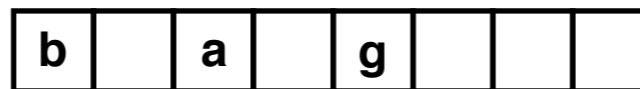


g requested

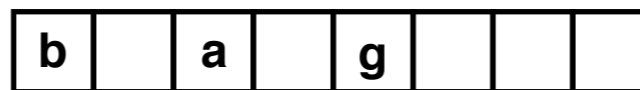


g brought in

b brought in



b requested



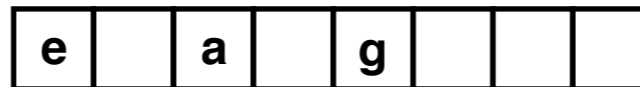
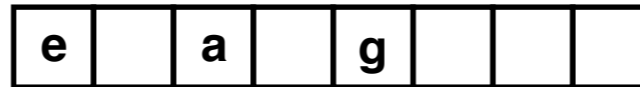
Optimality of FF

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

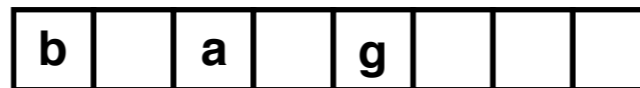
Sequence of requests



g requested



b requested



Cache changes

g brought in

b brought in

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 - -
FF	x - x - -
OPT'	x - x - -

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.

OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	

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.

OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

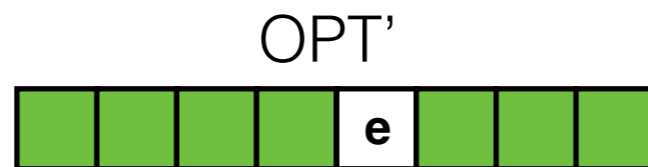
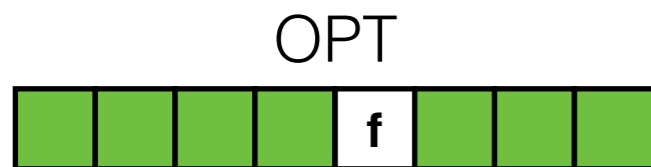
Optimality of FF

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.

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.

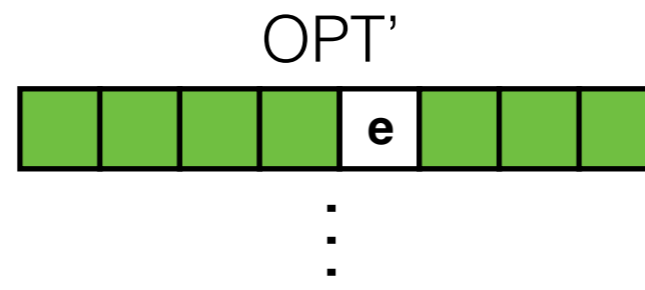
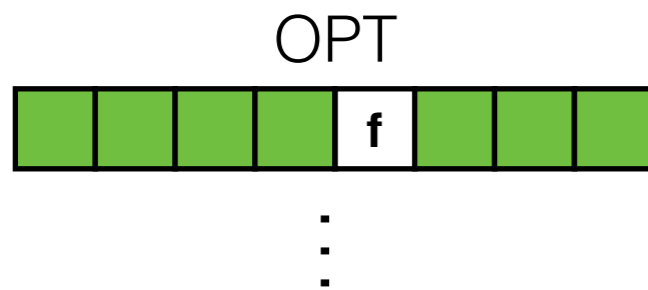
Main fact: if reduced OPT & FF are consistent on first k choices, then there exists a reduced OPT' that is consistent with FF on the first $k+1$ choices.

OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

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.

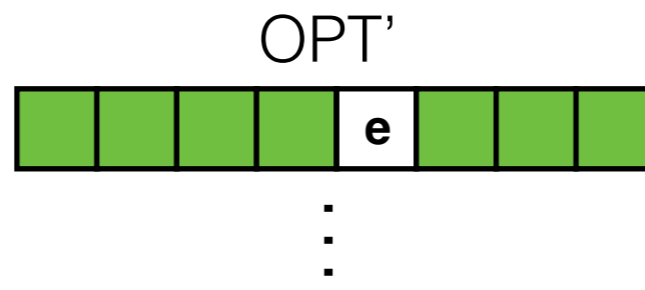
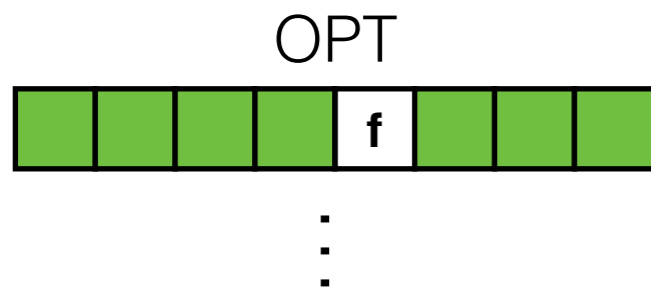
OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

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.

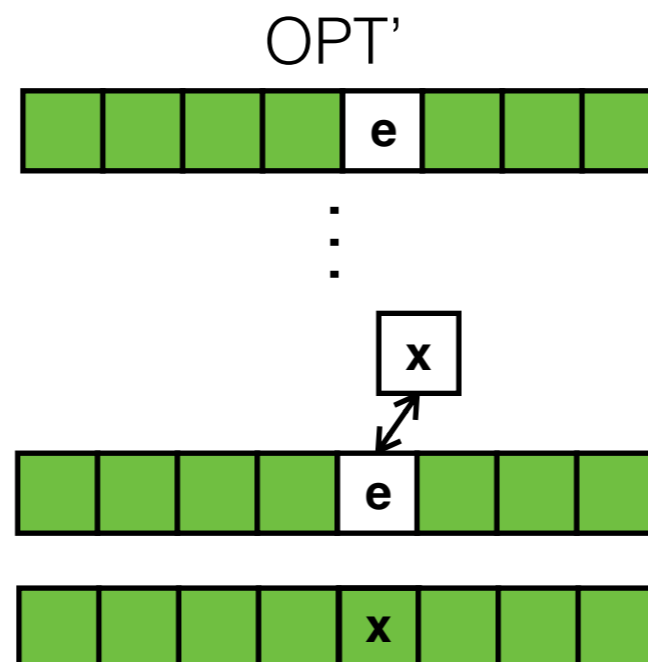
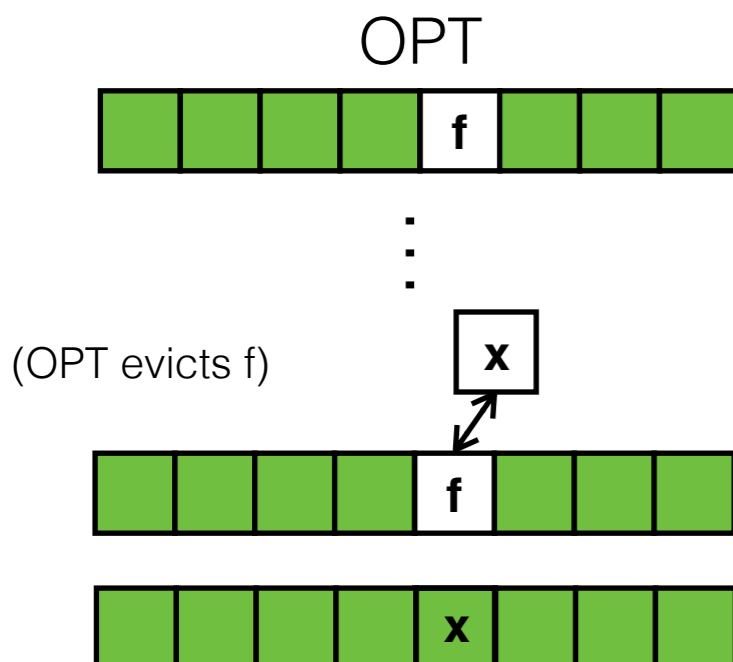
OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

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.

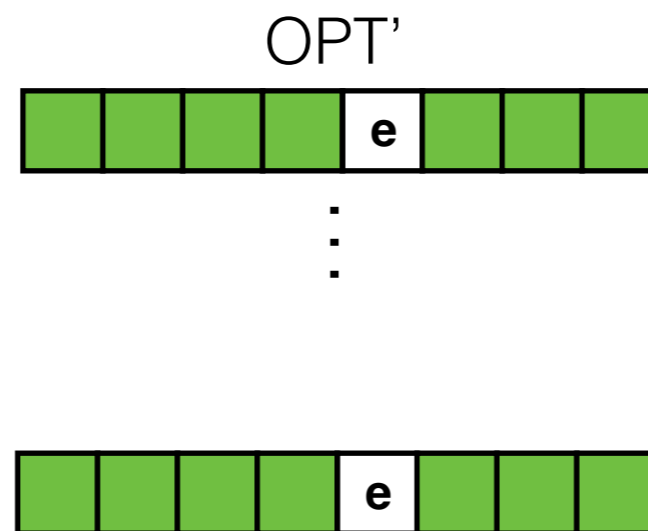
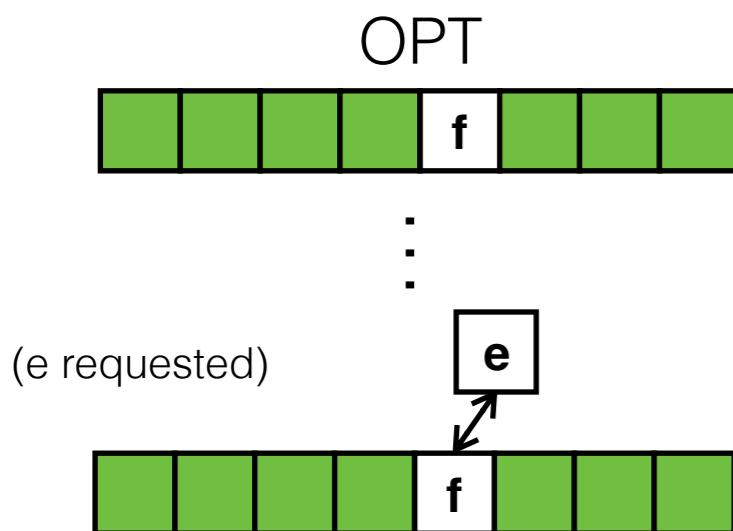
OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

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.

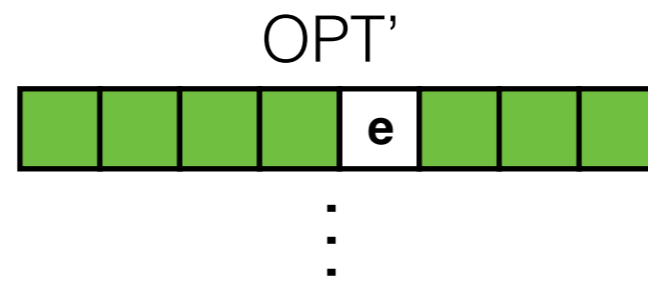
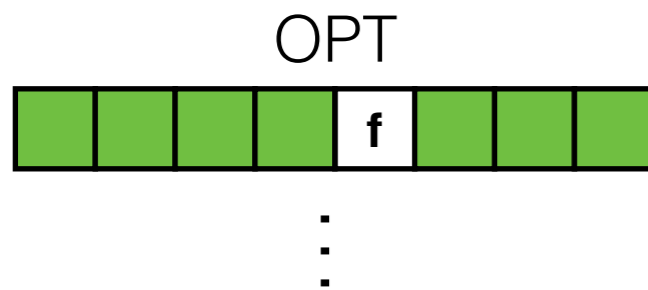
OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

Cache resolution:

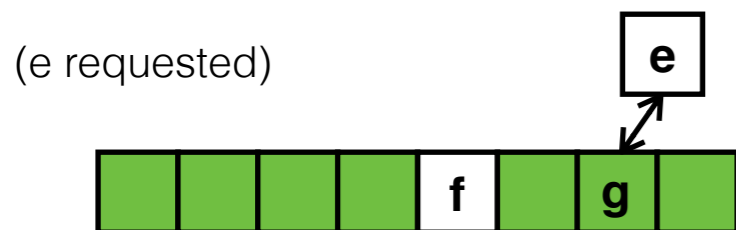
Case 1: e never requested again. Then f is never requested again, so resolution is trivial

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 $g \neq f$,



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.

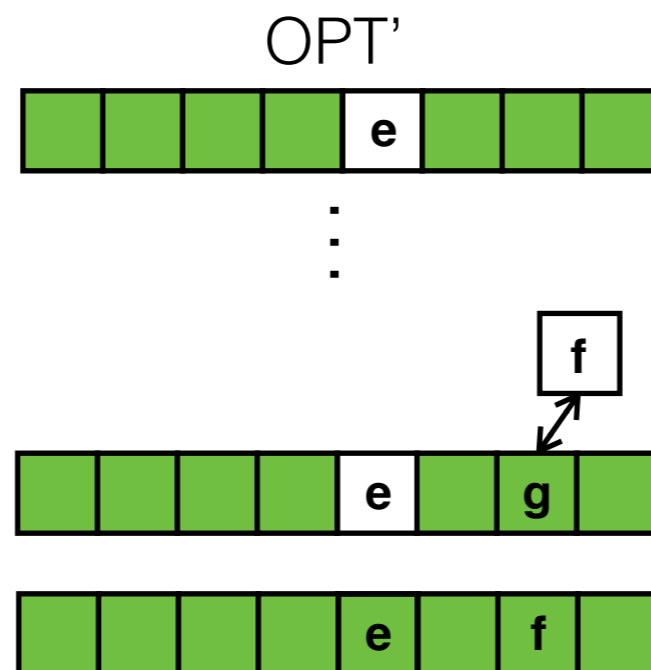
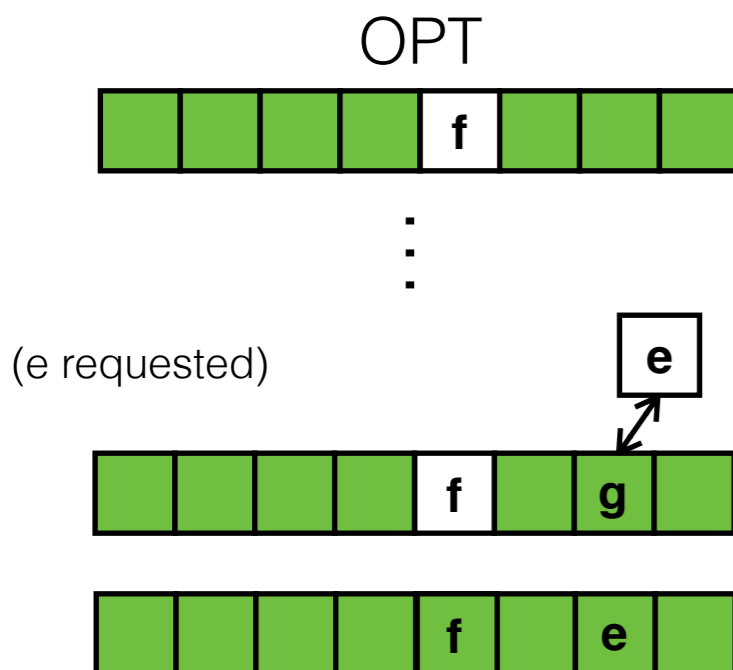
OPT	X - X - -	Xevict e
FF	X - X - -	Xevict f
OPT'	X - X - -	Xevict f

choice
 $k+1$

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 $g \neq f$, OPT' evicts g to bring in f.

OPT' not reduced!