Análise de Algoritmos

Estes slides são adaptações de slides

do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.

Cache

Modelo: itens do mesmo tamanho

Memória cache de tamanho k (ítens)

Sequência de requisições:

$$S : d_1, d_2, \ldots, d_m$$

Cache hit: acerto – item está no cache

Cache miss: falha – item não está no cache

Se o cache está cheio e entra um novo item, outro tem que sair (é despejado).

Políticas de despejo

Política de despejo (*replacement policy*):
algoritmo que decide quando trazer um item para o cache e que item despejar.

Política preguiçosa (*Belady's policy*):
um item só é trazido para o cache se ele foi requisitado.

Política FF (*farthest first*): despeja o elemento que será acessado novamente no futuro mais distante.

Vimos na aula passada que a política FF é ótima:
provoca o menor número possível de falhas.

Argumento do Lucas

Políticas online

LRU/MRU – least/most recently used

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Fase:

Desmarque todos os itens de C.

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Fase:

Desmarque todos os itens de C.
Seja d a próxima requisição.
Marque d.
Se $d \in C$ então atenda d.

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Fase:

Desmarque todos os itens de C.
Seja d a próxima requisição.
Marque d.
Se $d \in C$ então atenda d.
Se $d \notin C$ então
  Se C está todo marcado então
    antes de atender d, comece nova fase.

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Fase:

Desmarque todos os itens de C.
Seja d a próxima requisição.
Marque d.
Se $d \in C$ então atenda d.
Se $d \notin C$ então
  Se C está todo marcado então
    antes de atender d, comece nova fase.
  senão despeje de C um item desmarcado e atenda d.

Políticas online

LRU/MRU – least/most recently used

Algoritmo de marcação por fases:

Cada item do cache C está marcado ou desmarcado.

Fase:

Desmarque todos os itens de C.
Seja d a próxima requisição.
Marque d.
Se $d \in C$ então atenda d.
Se $d \notin C$ então
  Se C está todo marcado então
    antes de atender d, comece nova fase.
  senão despeje de C um item desmarcado e atenda d.

LRU é um algoritmo de marcação.

Análise de algoritmos de marcação

Trecho de $\geq k+1$ requisições sucessivas:

bom se tem $\geq k+1$ requisições distintas

ruim caso contrário.

Análise de algoritmos de marcação

Trecho de $\geq k+1$ requisições sucessivas:
bom se tem $\geq k+1$ requisições distintas
ruim caso contrário.

Em um trecho bom, ocorre pelo menos uma falha em qualquer política.

Cada fase termina no fim de um trecho bom de requisições.

Análise de algoritmos de marcação

Trecho de $\geq k+1$ requisições sucessivas:
bom se tem $\geq k+1$ requisições distintas
ruim caso contrário.

Em um trecho bom, ocorre pelo menos uma falha em qualquer política.

Cada fase termina no fim de um trecho bom de requisições.

Seja S uma sequência de requisições, e
$f(S)$ o número mínimo de falhas numa política para S.

Então $f(S) \geq r-1$, onde r é o número de fases.

Análise de algoritmos de marcação

Trecho de $\geq k+1$ requisições sucessivas:
bom se tem $\geq k+1$ requisições distintas
ruim caso contrário.

Em um trecho bom, ocorre pelo menos uma falha em qualquer política.

Cada fase termina no fim de um trecho bom de requisições.

Seja S uma sequência de requisições, e
$f(S)$ o número mínimo de falhas numa política para S.

Então $f(S) \geq r-1$, onde r é o número de fases.

Algoritmo de marcação: no máximo k falhas por fase.
Logo o número de falhas nele é no máximo $rk \leq kf(S)+k$.

Versão aleatorizada

Algoritmo de marcação aleatorizado:
sorteie um item desmarcado uniformemente para despejar.

Versão aleatorizada

Algoritmo de marcação aleatorizado:
sorteie um item desmarcado uniformemente para despejar.

Análise:

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.

Versão aleatorizada

Algoritmo de marcação aleatorizado:
sorteie um item desmarcado uniformemente para despejar.

Análise:

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.

c_1: número de itens fora do cache inicial na fase 1.
c_j: número de itens frescos na fase j, para $j \geq 2$.

Versão aleatorizada

Algoritmo de marcação aleatorizado:
sorteie um item desmarcado uniformemente para despejar.

Análise:

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.

c_1: número de itens fora do cache inicial na fase 1.
c_j: número de itens frescos na fase j, para $j \geq 2$.

$f_j(S)$: número de falhas da política ótima na fase j.

$$f(S) = \sum_{j=1}^{r} f_j(S)$$

Análise

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.
$f_j(S)$: número de falhas da política ótima na fase j.
c_j: número de itens frescos na fase j.
$f(S) = \sum_{j=1}^{r} f_j(S)$.

Análise

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.
$f_j(S)$: número de falhas da política ótima na fase j.
c_j: número de itens frescos na fase j.
$f(S) = \sum_{j=1}^{r} f_j(S)$.

Olhando fases j e $j+1$: $f_j(S) + f_{j+1}(S) \geq c_{j+1}$.

Análise

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.
$f_j(S)$: número de falhas da política ótima na fase j.
c_j: número de itens frescos na fase j.
$f(S) = \sum_{j=1}^{r} f_j(S)$.

Olhando fases j e $j+1$: $f_j(S) + f_{j+1}(S) \geq c_{j+1}$.

Logo

$$\sum_{j=1}^{r} c_j \leq \sum_{j=0}^{r-1} (f_j(S) + f_{j+1}(S)) \leq 2\, f(S).$$

Análise

Fase j:
Item desmarcado é fresco se não foi marcado na fase $j-1$.
Item desmarcado é amanhecido caso contrário.
$f_j(S)$: número de falhas da política ótima na fase j.
c_j: número de itens frescos na fase j.
$f(S) = \sum_{j=1}^{r} f_j(S)$.

Olhando fases j e $j+1$: $f_j(S) + f_{j+1}(S) \geq c_{j+1}$.

Logo

$$\sum_{j=1}^{r} c_j \leq \sum_{j=0}^{r-1} (f_j(S) + f_{j+1}(S)) \leq 2\, f(S).$$

Ou seja, $f(S) \geq (\sum_{j=1}^{r} c_j)/2$.

Análise do algoritmo aleatorizado

X_j: número de falhas na fase j

$X = \sum_j X_j$

Análise do algoritmo aleatorizado

X_j: número de falhas na fase j

$X = \sum_j X_j$

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

Análise do algoritmo aleatorizado

X_j: número de falhas na fase j

$X = \sum_j X_j$

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

Requisição do i-ésimo amanhecido nesta fase:
Cache tem $c \leq c_j$ frescos
$i - 1$ amanhecidos marcados
$k - c - i + 1$ amanhecidos desmarcados

Análise do algoritmo aleatorizado

X_j: número de falhas na fase j

$X = \sum_j X_j$

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

Requisição do i-ésimo amanhecido nesta fase:
Cache tem $c \leq c_j$ frescos
$i - 1$ amanhecidos marcados
$k - c - i + 1$ amanhecidos desmarcados

Número de amanhecidos fora do cache é c.
Número de amanhecidos desmarcados: $k - i + 1$.

Análise do algoritmo aleatorizado

X_j: número de falhas na fase j

$X = \sum_j X_j$

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

Requisição do i-ésimo amanhecido nesta fase:
Cache tem $c \leq c_j$ frescos
$i - 1$ amanhecidos marcados
$k - c - i + 1$ amanhecidos desmarcados

Número de amanhecidos fora do cache é c.
Número de amanhecidos desmarcados: $k - i + 1$.

$$\Pr[\text{falha no } i\text{-ésimo amanhecido}] = \frac{c}{k - i + 1}$$

Análise do algoritmo aleatorizado

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

$$\Pr[\text{falha no } i\text{-ésimo amanhecido}] = \frac{c}{k - i + 1}$$

Análise do algoritmo aleatorizado

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

$$\Pr[\text{falha no } i\text{-ésimo amanhecido}] \;=\; \frac{c}{k-i+1}$$

Logo

$$\mathrm{E}[X_j] \;\le\; c_j + \sum_{i=1}^{k-c_j} \frac{c_j}{k-i+1} \;=\; c_j(1 + H_k - H_{c_j}) \;\le\; H_k\, c_j.$$

Análise do algoritmo aleatorizado

$X_j = c_j$ falhas pelos itens frescos
$+$ falhas pelos amanhecidos.

$$\Pr[\text{falha no } i\text{-ésimo amanhecido}] \;=\; \frac{c}{k-i+1}$$

Logo

$$\mathrm{E}[X_j] \;\le\; c_j + \sum_{i=1}^{k-c_j} \frac{c_j}{k-i+1} \;=\; c_j(1 + H_k - H_{c_j}) \;\le\; H_k\, c_j.$$

Como $X = \sum_j X_j$, temos que

$$\mathrm{E}[X] \;=\; \sum_j \mathrm{E}[X_j] \;\le\; H_k \sum_j c_j \;\le\; 2\, H_k\, f(S) \;=\; \mathrm{O}(\lg k)\, f(S).$$