Algoritmer og Datastrukturer

Prioritetskøer med Afskæring

Rajamani Sundar, *Worst-case data structures for the priority queue with attrition*, Information Processing Letters, 31(2), 69-75, 1989, DOI: 10.1016/0020-0190(89)90071-9

Information Processing Letters 31 (1989) 69–75
North-Holland

26 April 1989

WORST-CASE DATA STRUCTURES FOR THE PRIORITY QUEUE WITH ATTRITION

Rajamani SUNDAR *

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, U.S.A.

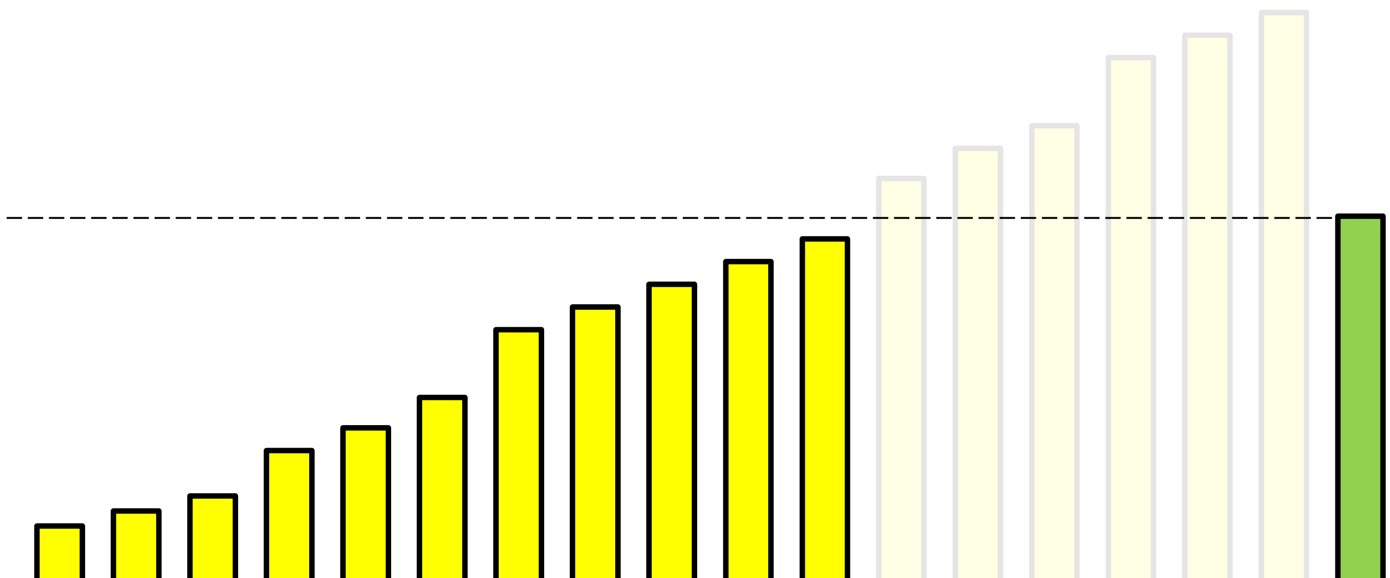
Communicated by David Gries
Received 15 June 1988
Revised 14 November 1988

We describe three data structures for the priority queue with attrition (or PQA) that perform each PQA operation in O(1) worst-case time. Previous implementations of the PQA required O(1) amortized time per operation.

Keywords: Priority queue with attrition, worst-case

Operationer

Create	$S := \emptyset$
Insert(*x*)	$S := \{\ x\ \} \cup \{\ y \in S \mid y < x\ \}$
Deletemin	$m := \min(S);\ S := S \setminus \{\ m\ \};$ return m



Insert(4) i {1,3,5,6,9} ?

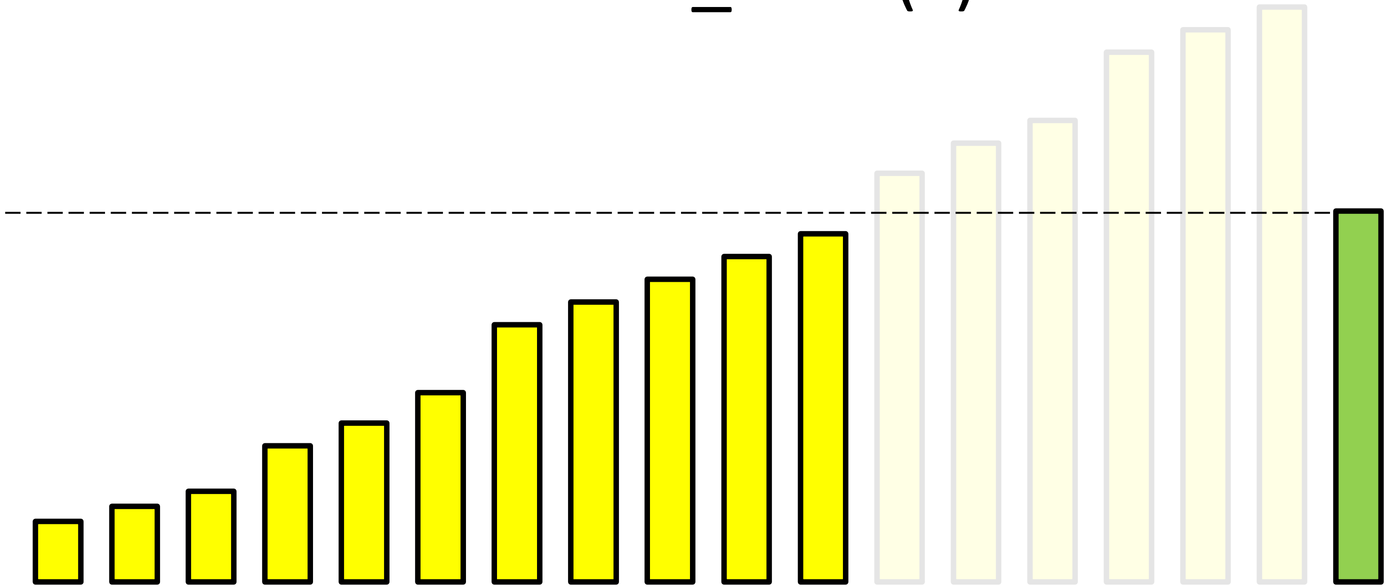
a) {4,5,6,9}
b) {1,3,4,5,6,9}
c) {1,3,4}
d) ved ikke

Løsning: Sorteret Liste

Create S := ()

Insert(*x*) while (|S|>0 and tail(S)≥x) remove_tail(S)
insert_tail(x)

Deletemin return remove_head(S)



Løsning: Sorteret Liste

Create $S := ()$

Insert(*x*) while $(|S|>0$ and $\text{tail}(S) \geq x)$ remove_tail(S)
insert_tail(x)

Deletemin return remove_head(S)

Sætning
Create, Insert og DeleteMin tager **amortiseret** O(1) tid

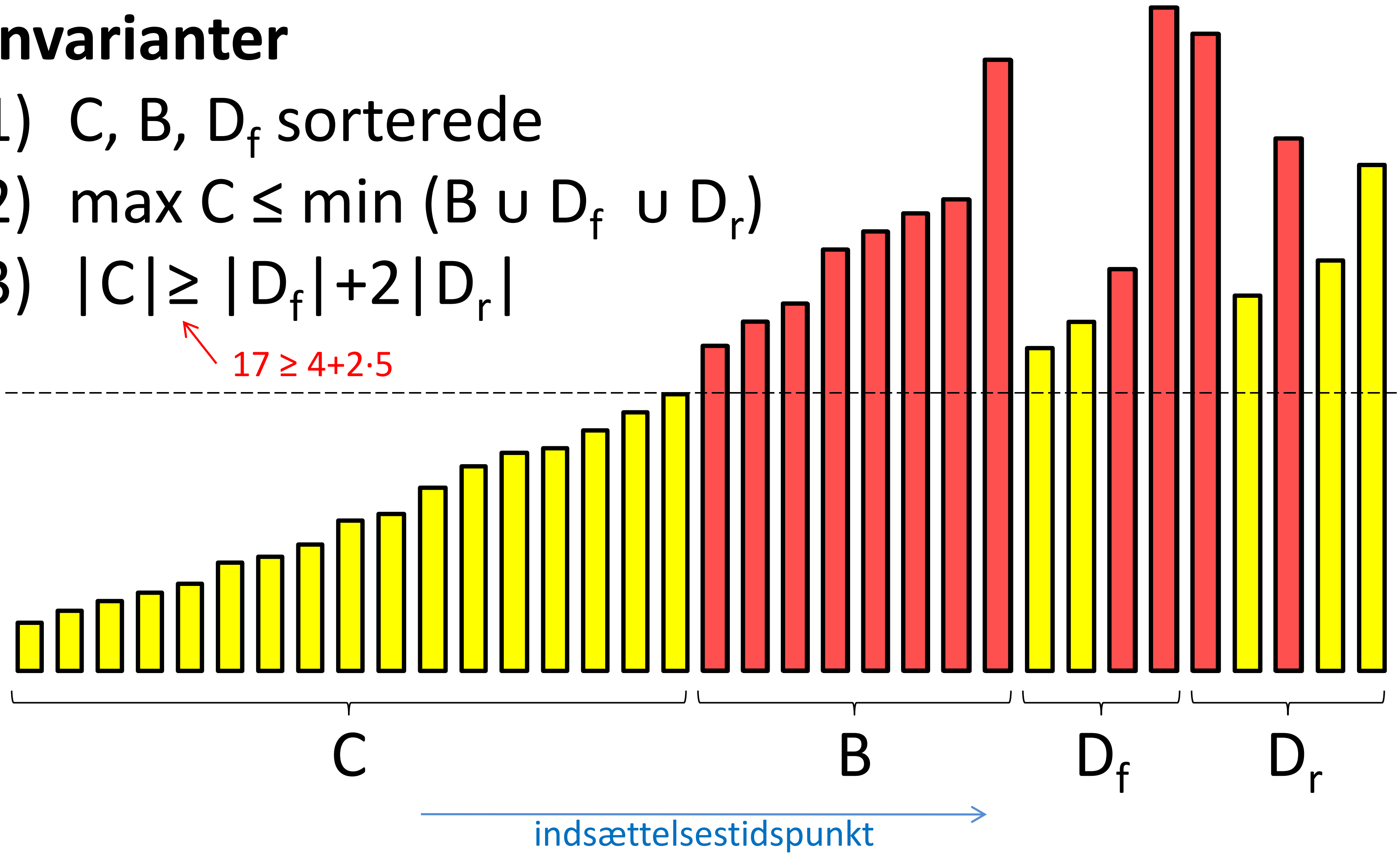
Bevis: $\Phi(S) = |S|$. □

Worst-Case $O(1)$

Løsning: 4 Lister

Invarianter

1) C, B, D_f sorterede
2) max C ≤ min (B ∪ D_f ∪ D_r)
3) $|C| \geq |D_f| + 2|D_r|$



Hvilken mængde repræsenteres ved:

$$(1,2,3,4,5,6)(7,10,14)(8,11)(9,8)$$

C B D_f D_r

a) {1,2,3,4,5,6,7,8,8,9,10,11,14}
b) {1,2,3,4,5,6,7,8,8,9}
c) {1,2,3,4,5,6,7,8,9}
d) {1,2,3,4,5,6,7,8}
e) {1,2,3,4,5,6,7,8,8}
f) ved ikke

Invarianter

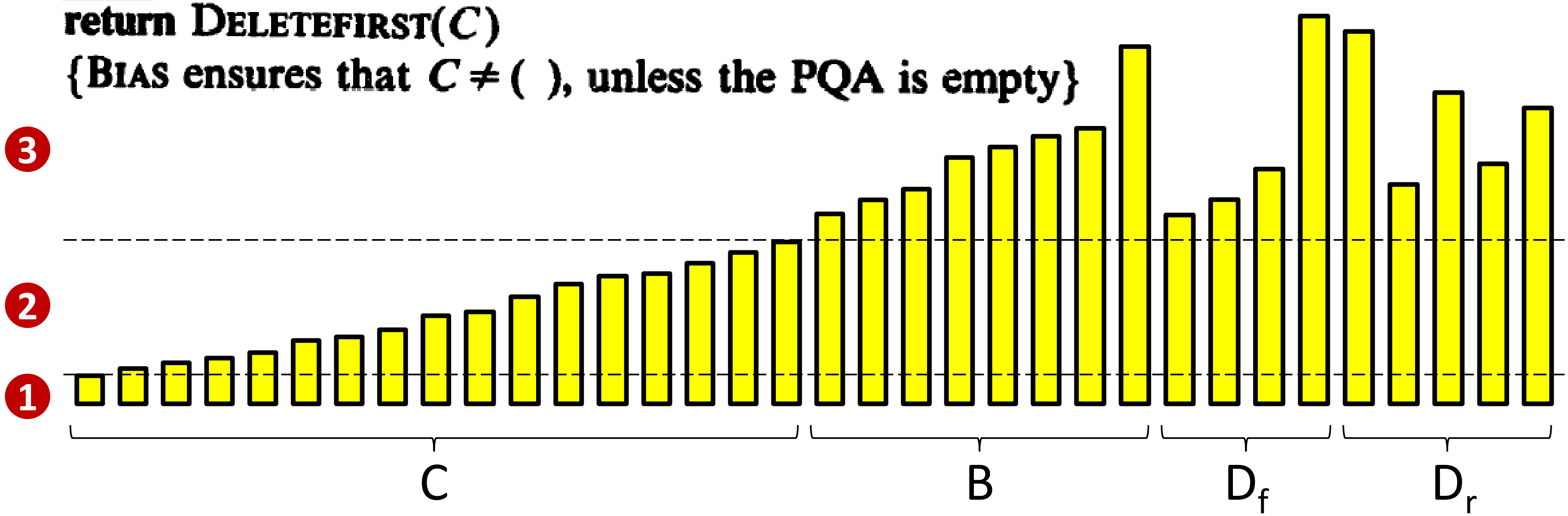
1) C, B, D_f sorterede
2) max C ≤ min (B ∪ D_f ∪ D_r)
3) |C| ≥ $|D_f|$ + 2$|D_r|$

≥ "+1" BIAS

CREATEPQA ≡
    C, B, D_f, D_r := ( ), ( ), ( ), ( )

INSERT(x) ≡
    if $C \neq (\ )$ **cand first**(C) ≥ x **then**
        {Delete all existing items; add x to C}
❶ C, B, D_f, D_r := (x), ( ), ( ), ( )
    else if $C \neq (\ )$ **cand last**(C) ≥ x **then**
        {Empty B, D_f, and D_r; push back rest(C) into B; add x to D_f}
❷ C, B, D_f, D_r := (first(C)), rest(C), (x), ( )
❸   **else** $D_r := D_r \| (x)$; BIAS; BIAS

DELETEMIN ≡
    BIAS;
    return DELETEFIRST(C)
    {BIAS ensures that $C \neq (\ )$, unless the PQA is empty}

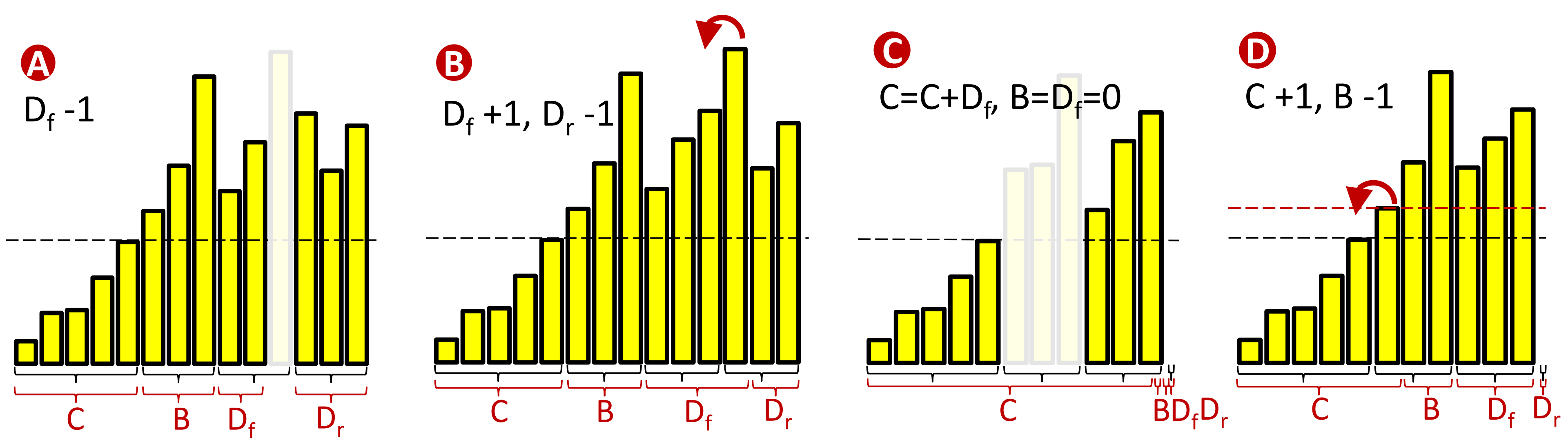


Invarianter

1) C, B, D_f sorterede
2) max C ≤ min (B ∪ D_f ∪ D_r)
3) |C| ≥ $|D_f| + 2|D_r|$

≥ "+1" BIAS

BIAS ≡
- **if** $D_r \neq (\ )$ **then**
  - {Clean-up step}
  - **if** $D_f \neq (\ )$ **cand** last(D_f) ≥ first(D_r) **then**
    - (A) DELETELAST(D_f) {decrease $|D_f|$}
  - (B) **else** PASS(D_r, D_f) {decrease $|D_r|$; increase $|D_f|$}
- **else if** $D_f \neq (\ )$ **cand** ($B = (\ )$ **cor** first(B) ≥ first(D_f)) **then**
  - (C) $D_f, B, C := (\ ), (\ ), C \,\|\, D_f$ {decrease $|D_f|$; increase $|C|$}
- (D) **else if** $B \neq (\ )$ **then** PASS(B, C) {increase $|C|$}
- {else $B = D_f = D_r = (\ )$}



Resultatet af **insert(7)** på:

(1,2,3,4,5,6)(7,10,14)(8,11)(9,8)

C B D_f D_r

a) (1,2,3,4,5,6)(7,10,14)(8,11)(9,8,7)
b) (1,2,3,4,5,6)(7,10,14)(8,9)(8,7)
c) (1,2,3,4,5,6)(7,10,14)(8)(9,8,7)
d) (1,2,3,4,5,6,7)(10,14)(8)(9,8,7)
e) ved ikke

Sætning

Create, Insert og DeleteMin tager **worst-case** O(1) tid

Invarianter

1) C, B, D_f sorterede
2) max C ≤ min (B ∪ D_f ∪ D_r)
3) |C| ≥ |D_f|+2|D_r|