Counting Inversions, Offline Orthogonal Range Counting, and Related Problems

Timothy M. Chan Mihai Pătraşcu

University of Waterloo

at&t

SODA'10

Counting Inversions

Inversion Counting: given permutation π, output $\mathit{Inv}(\pi)$.

Counting Inversions

Inversion Counting: given permutation π, output $Inv(\pi)$.

Partial sums: Maintain $A[1..n]$ under:

UPDATE(k, x): $A[k] = x$.

SUM(k): output $\sum_{i=1}^{k} A[i]$.

Counting Inversions

Inversion Counting: given permutation π, output $Inv(\pi)$.

Partial sums: Maintain $A[1\,..\,n]$ under:

$\textsc{Update}(k, x)$: $A[k] = x$.

$\textsc{Sum}(k)$: output $\sum_{i=1}^{k} A[i]$.

How to count inversions:

$A[1\,..\,n] = 0$
$inv = 0$
for $k = 1$ **to** n
    $inv = inv + (k-1) - \textsc{Sum}(\pi(k))$
    $\textsc{Update}(\pi(k), 1)$

Complexity

Partial sums with $A[i] \in \{0, 1\}$:

[Dietz'89] $t_u = t_q = O(\frac{\lg n}{\lg \lg n})$

[Fredman-Saks'89] $t_q = \Omega(\frac{\lg n}{\lg \lg n})$ for any $t_u \leq \lg^{O(1)} n$

Counting inversions:

[Andersson-Petersson'95] Can you beat $O(n \frac{\lg n}{\lg \lg n})$?

Complexity

Partial sums with $A[i] \in \{0, 1\}$:

[Dietz'89] $t_u = t_q = O(\frac{\lg n}{\lg \lg n})$

[Fredman-Saks'89] $t_q = \Omega(\frac{\lg n}{\lg \lg n})$ for any $t_u \leq \lg^{O(1)} n$

Counting inversions:

[Andersson-Petersson'95] Can you beat $O(n\frac{\lg n}{\lg \lg n})$?

[Chan-P.'07] "This problem is no easier than counting inversions in a permutation, so the best bound one would hope to get is $O(n\frac{\lg n}{\lg \lg n})$."

Complexity

Partial sums with $A[i] \in \{0, 1\}$:

[Dietz'89] $t_u = t_q = O(\frac{\lg n}{\lg \lg n})$

[Fredman-Saks'89] $t_q = \Omega(\frac{\lg n}{\lg \lg n})$ for any $t_u \leq \lg^{O(1)} n$

Counting inversions:

[Andersson-Petersson'95] Can you beat $O(n \frac{\lg n}{\lg \lg n})$?

[Chan-P.'07] "This problem is no easier than counting inversions in a permutation, so the best bound one would hope to get is $O(n \frac{\lg n}{\lg \lg n})$."

...Fortunately, we grow smarter every year ...

NEW: $O(n\sqrt{\lg n})$.

Complexity

Partial sums with $A[i] \in \{0, 1\}$:

[Dietz'89] $t_u = t_q = O(\frac{\lg n}{\lg \lg n})$

[Fredman-Saks'89] $t_q = \Omega(\frac{\lg n}{\lg \lg n})$ for any $t_u \le \lg^{O(1)} n$

NEW: $t_q = O(\frac{\lg n}{\lg \lg n})$, $t_u = O(\lg^{0.5+\varepsilon} n)$.

Counting inversions:

[Andersson-Petersson'95] Can you beat $O(n \frac{\lg n}{\lg \lg n})$?

[Chan-P.'07] "This problem is no easier than counting inversions in a permutation, so the best bound one would hope to get is $O(n \frac{\lg n}{\lg \lg n})$."

... Fortunately, we grow smarter every year ...

NEW: $O(n \sqrt{\lg n})$.

External Memory

Well-known fact: in external memory, online $\neq$ offline

predecessor search: B-trees, $O(\frac{\lg n}{\lg B})$;

sorting: $O(\frac{n \lg n}{B})$. [NB: Ignore cache for this talk.]

External Memory

Well-known fact: in external memory, online $\neq$ offline

predecessor search: B-trees, $O(\frac{\lg n}{\lg B})$;

sorting: $O(\frac{n \lg n}{B})$.

[NB: Ignore cache for this talk.]

How about counting inversions?

n L-bit numbers $\Rightarrow$ $O(\frac{nL}{B})$ via recursive radix-sort

External Memory

Well-known fact: in external memory, online $\neq$ offline

predecessor search: B-trees, $O(\frac{\lg n}{\lg B})$;

sorting: $O(\frac{n \lg n}{B})$.

[NB: Ignore cache for this talk.]

How about counting inversions?

n L-bit numbers $\Rightarrow$ $O(\frac{nL}{B})$ via recursive radix-sort

```
Count(A)

zeros = ones = {}, inv = 0
for k = 1 to |A|
      if MSB(A[i]) == 0
            inv = inv + |ones|
            APPEND(zeros, DISCARDMSB(A[i]))
      else APPEND(ones, DISCARDMSB(A[i]))
return inv + COUNT(zeros) + COUNT(ones)
```

Complexity $\approx$ L list traversals.

Now Remember van Emde Boas

Integer	$\Rightarrow$	String
$\lg n$ bits	$\Rightarrow$	$\sqrt{\lg n}$ characters $\times \sqrt{\lg n}$ bits each

Now Remember van Emde Boas

Integer	$\Rightarrow$	String
$\lg n$ bits	$\Rightarrow$	$\sqrt{\lg n}$ characters $\times \sqrt{\lg n}$ bits each

Build a trie: $\Sigma = \left[2^{\sqrt{\lg n}}\right]$, height $\log_\Sigma n = \sqrt{\lg n}$.

$Inv(\pi) = \sum_{v \in \text{Nodes}} Inv(\text{characters at } v)$

Now Remember van Emde Boas

Integer	$\Rightarrow$	String
$\lg n$ bits	$\Rightarrow$	$\sqrt{\lg n}$ characters $\times \sqrt{\lg n}$ bits each

Build a trie: $\Sigma = \left[2^{\sqrt{\lg n}}\right]$, height $\log_\Sigma n = \sqrt{\lg n}$.

$Inv(\pi) = \sum_{v \in \text{Nodes}} Inv(\text{characters at } v)$

The subproblem at each node

Items have $L = \sqrt{\lg n}$ bits

Words = $O(\lg n)$ bits
$\Rightarrow B = O(\sqrt{\lg n})$ items can be manipulated in constant time.†

† Using tabulation.

$O(\frac{nL}{B}) = O(n)$ time.

List of Results

Improved algorithms:

offline range counting: $O\left(n\left(\frac{\lg n}{\lg\lg n}\right)^{d-1}\right) \longrightarrow O\left(n \lg^{d-2+\frac{1}{d}} n\right)$

offline $\{0,1\}$ partial sums: $O\left(n\frac{\lg n}{\lg\lg n}\right) \longrightarrow O\left(n\sqrt{\lg n}\cdot\sqrt[4]{\lg\lg n}\right)$

maximum depth in rectangle arrangement: $O(n\sqrt{\lg n})$

slope selection: $O(n\sqrt{\lg n})$

List of Results

Improved algorithms:

offline range counting: $O\left(n\left(\frac{\lg n}{\lg\lg n}\right)^{d-1}\right) \longrightarrow O\left(n \lg^{d-2+\frac{1}{d}} n\right)$

offline $\{0,1\}$ partial sums: $O(n\frac{\lg n}{\lg\lg n}) \longrightarrow O(n\sqrt{\lg n}\cdot\sqrt[4]{\lg\lg n})$

maximum depth in rectangle arrangement: $O(n\sqrt{\lg n})$

slope selection: $O(n\sqrt{\lg n})$

Static 2D range counting:

- space $O(n)$, query time $O(\frac{\lg n}{\lg\lg n})$ [OPTIMAL]
- construction time $O(n\sqrt{\lg n})$ [NEW]

List of Results

Improved algorithms:

offline range counting: $O\left(n\left(\frac{\lg n}{\lg\lg n}\right)^{d-1}\right) \longrightarrow O\left(n \lg^{d-2+\frac{1}{d}} n\right)$

offline $\{0,1\}$ partial sums: $O(n\frac{\lg n}{\lg\lg n}) \longrightarrow O(n\sqrt{\lg n} \cdot \sqrt[4]{\lg\lg n})$

maximum depth in rectangle arrangement: $O(n\sqrt{\lg n})$

slope selection: $O(n\sqrt{\lg n})$

Static 2D range counting:

- space $O(n)$, query time $O(\frac{\lg n}{\lg\lg n})$ [OPTIMAL]
- construction time $O(n\sqrt{\lg n})$ [NEW]

Online $\{0,1\}$ partial sums:

- query time $O(\frac{\lg n}{\lg\lg n})$ [OPTIMAL]
- update time $O(\lg^{0.5+\varepsilon} n)$ [NEW]

Thank you!

THE END