Sorting, Searching, and Amortized Analysis

Hengfeng Wei

hfwei@nju.edu.cn

April 24, 2018

Maximal-sum Subarray (Problem 3.7)

- Array $A[1 \cdots n], a_i >=< 0$
- To find (the sum of) an MS in A

$$A[-2, 1, -3, \boxed{4, -1, 2, 1}, -5, 4]$$

MSS$[i]$: the sum of the MS (MS$[i]$) in $A[1 \cdots i]$

$\mathsf{MSS}[i]$: the sum of the MS ($\mathsf{MS}[i]$) in $A[1 \cdots i]$

$$\mathsf{mss} = \mathsf{MSS}[n]$$

MSS$[i]$: the sum of the MS (MS$[i]$) in $A[1 \cdots i]$

$$\text{mss} = \text{MSS}[n]$$

Q : Is $a_i \in$ MS$[i]$?

MSS$[i]$: the sum of the MS (MS$[i]$) in $A[1 \cdots i]$

$$\text{mss} = \text{MSS}[n]$$

Q : Is $a_i \in$ MS$[i]$?

$$\text{MSS}[i] = \max\{\text{MSS}[i-1], ???\}$$

MSS$[i]$: the sum of the MS *ending with* a_i or 0

MSS$[i]$: the sum of the MS *ending with* a_i or 0

$$\boxed{\mathsf{mss} = \max_{1 \le i \le n} \mathsf{MSS}[i]}$$

Q : where does the MS$[i]$ start?

MSS$[i]$: the sum of the MS *ending with* a_i or 0

$$\boxed{\text{mss} = \max_{1 \le i \le n} \text{MSS}[i]}$$

Q : where does the MS$[i]$ start?

$$\boxed{\text{MSS}[i] = \max\left\{\text{MSS}[i-1] + a_i, 0\right\}}$$

MSS$[i]$: the sum of the MS *ending with* a_i or 0

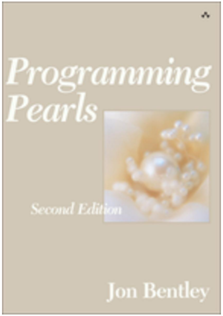
$$\boxed{\text{mss} = \max_{1 \le i \le n} \text{MSS}[i]}$$

Q : where does the MS$[i]$ start?

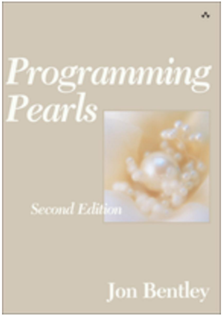
$$\boxed{\text{MSS}[i] = \max\left\{\text{MSS}[i-1] + a_i, 0\right\}}$$

$$\text{MSS}[0] = 0$$

```
procedure MSS(A[1 ··· n])
    MSS[0] ← 0
    for i ← 1 to n do
        MSS[i] ← max {MSS[i − 1] + A[i], 0}
    return max_{1≤i≤n} MSS[i]
```

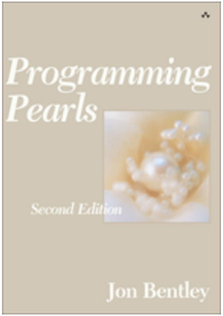


Ulf Grenander $O(n^3) \implies O(n^2)$



Ulf Grenander $O(n^3) \implies O(n^2)$

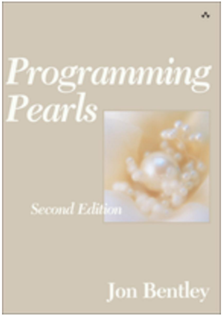
Michael Shamos $O(n \log n)$, onenight



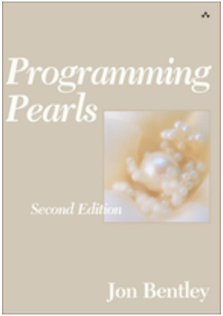
Ulf Grenander $O(n^3) \implies O(n^2)$

Michael Shamos $O(n \log n)$, onenight

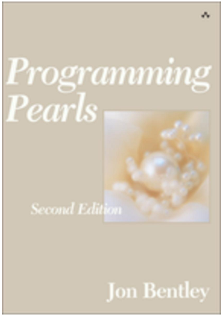
Jon Bentley Conjecture: $\Omega(n \log n)$



Ulf Grenander	$O(n^3) \implies O(n^2)$
Michael Shamos	$O(n \log n)$, onenight
Jon Bentley	Conjecture: $\Omega(n \log n)$
Michael Shamos	Carnegie Mellon seminar



Ulf Grenander	$O(n^3) \implies O(n^2)$
Michael Shamos	$O(n \log n)$, onenight
Jon Bentley	Conjecture: $\Omega(n \log n)$
Michael Shamos	Carnegie Mellon seminar
Jay Kadane	$O(n)$,



Ulf Grenander	$O(n^3) \implies O(n^2)$
Michael Shamos	$O(n \log n)$, onenight
Jon Bentley	Conjecture: $\Omega(n \log n)$
Michael Shamos	Carnegie Mellon seminar
Jay Kadane	$O(n)$, ≤ 1 minute

Definition (K-sorting (Problem 6.8))

An array $A[1 \cdots n]$ is *k-sorted* if it can be divided into k blocks, each of size n/k (we assume that $n/k \in \mathbb{N}$), such that the elements in each block are larger than the elements in earlier blocks and smaller than elements in later blocks. The elements within each block need *not* be sorted.

$$n = 16,\ k = 4,\ \frac{n}{k} = 4$$

$$1,\ 2,\ 4,\ 3; \quad 7,\ 6,\ 8,\ 5; \quad 10,\ 11,\ 9,\ 12; \quad 15,\ 13,\ 16,\ 14$$

k-sorted

k-sorted

1-sorted

k-sorted

1-sorted $\to$ 2-sorted

k-sorted

1-sorted $\to$ 2-sorted $\to$ 4-sorted

k-sorted

$$1\text{-sorted} \to 2\text{-sorted} \to 4\text{-sorted} \to \cdots \to n\text{-sorted}$$

k-sorted

1-sorted $\rightarrow$ 2-sorted $\rightarrow$ 4-sorted $\rightarrow \cdots \rightarrow$ n-sorted

Quicksort (with median as pivot) stops after the $\log k$ recursions.

k-sorted

1-sorted $\rightarrow$ 2-sorted $\rightarrow$ 4-sorted $\rightarrow \cdots \rightarrow$ n-sorted

Quicksort (with median as pivot) stops after the $\log k$ recursions.

$$\boxed{\Theta(n \log k)}$$

$$\Omega(n \log k)$$

$$\Omega(n \log k)$$

$$L =$$

$$\Omega(n \log k)$$

$$L = \binom{n}{n/k, \ldots, n/k}$$

$$\Omega(n \log k)$$

$$L = \binom{n}{n/k, \ldots, n/k} = \frac{n!}{\left(\left(\frac{n}{k}\right)!\right)^k}$$

$$\Omega(n \log k)$$

$$L = \binom{n}{n/k, \ldots, n/k} = \frac{n!}{\left((\frac{n}{k})!\right)^k}$$

$$H \geq$$

$$\Omega(n \log k)$$

$$L = \binom{n}{n/k, \ldots, n/k} = \frac{n!}{\left(\left(\frac{n}{k}\right)!\right)^k}$$

$$H \geq \log \left(\frac{n!}{\left(\left(\frac{n}{k}\right)!\right)^k} \right)$$

$$\Omega(n \log k)$$

$$L = \binom{n}{n/k, \ldots, n/k} = \frac{n!}{\left((\frac{n}{k})!\right)^k}$$

$$H \geq \log \left(\frac{n!}{\left((\frac{n}{k})!\right)^k} \right)$$

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \implies \log n! \sim n \log n$$

Bolts and Nuts (Problem 6.9)

Bolts and Nuts (Problem 6.9)

Quicksort

Bolts and Nuts (Problem 6.9)

Quicksort

$$A(n) = O(n \log n)$$

Bolts and Nuts (Problem 6.9)

Quicksort

$$A(n) = O(n \log n)$$

In the worst case:

- "Matching Nuts and Bolts" by Alon *et al.*, $\Theta(n \log^4 n)$
- "Matching Nuts and Bolts Optimality" by Bradford, 1995, $\Theta(n \log n)$

$$\Omega(n \log n)$$

$$\Omega(n \log n)$$

$$3^H \geq L \geq n!$$

$$\Omega(n \log n)$$

$$3^H \geq L \geq n! \implies H \geq \log n! \implies H = \Omega(n \log n)$$

Repeated elements (Problem 2.12)

$$R[1 \dots n]$$

$$\mathsf{check}(R[i], R[j])$$

$$\# > \frac{n}{13}$$

Repeated elements (Problem 2.12)

$$R[1 \ldots n]$$

$$\text{check}(R[i], R[j])$$

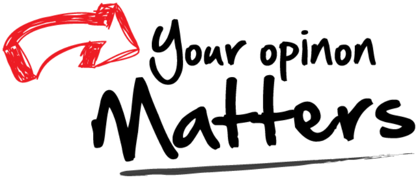
$$\# > \frac{n}{13}$$

$$\# > \frac{n}{k}$$

an $O(n \log k)$ algorithm
the lower bound $\Omega(n \log k)$

"Finding Repeated Elements" by Misra & Gries, 1982


Thank
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn