Asymptotics, Recurrences, and Divide and Conquer

Hengfeng Wei

hfwei@nju.edu.cn

March 29, 2017

Asymptotics, Recurrences, and Divide and Conquer

Integer multiplication (Problem 2.15)

Integer Multiplication

Multiplying two n-bit integers in $o(n^2)$ time. (Assuming $n = 2^k$.)

Column multiplication in $\Theta(n^2)$

Elementray operations:

- n-bit $+$ n-bit: $O(n)$
- 1-bit $\times$ 1-bit: $O(1)$
- n-bit shifted by 1-bit: $O(1)$

Integer multiplication (Problem 2.15)

Simple divide and conquer:

$$x = x_L : x_R = 2^{n/2} x_L + x_R$$
$$y = y_L : y_R = 2^{n/2} y_L + y_R$$

$$\begin{aligned} xy &= (2^{n/2} x_L + x_R)(2^{n/2} y_L + y_R) \\ &= 2^n x_L y_L + 2^{n/2}(x_L y_R + x_R y_L) + x_R y_R \end{aligned}$$

$$T(n) = 4T(n/2) + \Theta(n) = \Theta(n^2)$$

Integer multiplication (Problem 2.15)

A little history:

- Kolmogorov (1952) conjecture: $\Omega(n^2)$
- Kolmogorov (1960) seminar
- Karatsuba (*within a week*): $\Theta(n^{1.59})$
- "The Complexity of Computations" by Karatsuba, 1995

Integer multiplication (Problem 2.15)

Karatsuba algorithm:

$$T(n) = 3T(n/2) + \Theta(n) = \Theta(n^{\log_2 3}) = \Theta(n^{1.59})$$

Integer multiplication (Problem 2.15)

Karatsuba algorithm:

$$T(n) = 3T(n/2) + \Theta(n) = \Theta(n^{\log_2 3}) = \Theta(n^{1.59})$$

$$xy = 2^n x_L y_L + 2^{n/2}(x_L y_R + x_R y_L) + x_R y_R$$

$$\underbrace{(x_L + x_R)(y_L + y_R)}_{P_0} = \underbrace{x_L y_L}_{P_1} + (x_L y_R + x_R y_L) + \underbrace{x_R y_R}_{P_2}$$

$$xy = 2^n P_1 + 2^{n/2}(P_0 - P_1 - P_2) + P_2$$

Matrix multiplication (Problem 2.16)

Matrix multiplication

Multiplying two $n \times n$ matrices in $o(n^3)$ time. (Assuming $n = 2^k$.)

$$Z = X \times Y$$

$$Z_{ij}$$

$$T(n) = \Theta(n^2 \cdot n) = \Theta(n^3)$$

Elementrary operations:

- integer addition: $O(1)$
- integer multiplication: $O(1)$

Matrix multiplication (Problem 2.16)

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \qquad (A \ldots H \in \mathbb{R}^{n/2} \times \mathbb{R}^{n/2})$$

$$XY = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$$T(n) = 8T(n/2) + \Theta(n^2) = \Theta(n^3)$$

Matrix multiplication (Problem 2.16)

Strassen algorithm:

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\lg 7}) = \Theta(n^{2.808})$$

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$\begin{aligned}
P_1 &= A(F - H) \\
P_2 &= (A + B)H \\
P_3 &= (C + D)E \\
P_4 &= D(G - E) \\
P_5 &= (A + D)(E + H) \\
P_6 &= (B - D)(G + H) \\
P_7 &= (A - C)(E + F)
\end{aligned}$$

Matrix multiplication (Problem 2.16)

Strassen algorithm:

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\lg 7}) = \Theta(n^{2.808})$$

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$P_1 = A(F - H)$$
$$P_2 = (A + B)H$$
$$P_3 = (C + D)E$$
$$P_4 = D(G - E)$$
$$P_5 = (A + D)(E + H)$$
$$P_6 = (B - D)(G + H)$$
$$P_7 = (A - C)(E + F)$$

- Strassen (1969): $\Theta(n^{2.808})$
  "Gaussian Elimination is Not Optimal"

Matrix multiplication (Problem 2.16)

Strassen algorithm:

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\lg 7}) = \Theta(n^{2.808})$$

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$\begin{aligned}
P_1 &= A(F - H) \\
P_2 &= (A + B)H \\
P_3 &= (C + D)E \\
P_4 &= D(G - E) \\
P_5 &= (A + D)(E + H) \\
P_6 &= (B - D)(G + H) \\
P_7 &= (A - C)(E + F)
\end{aligned}$$

- Strassen (1969): $\Theta(n^{2.808})$
  "Gaussian Elimination is Not Optimal"
- (2014): $\Theta(n^{2.373})$

Matrix multiplication (Problem 2.16)

Strassen algorithm:

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\lg 7}) = \Theta(n^{2.808})$$

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$P_1 = A(F - H)$$
$$P_2 = (A + B)H$$
$$P_3 = (C + D)E$$
$$P_4 = D(G - E)$$
$$P_5 = (A + D)(E + H)$$
$$P_6 = (B - D)(G + H)$$
$$P_7 = (A - C)(E + F)$$

- Strassen (1969): $\Theta(n^{2.808})$
  "Gaussian Elimination is Not Optimal"
- (2014): $\Theta(n^{2.373})$
- Known lower bound: $\Omega(n^2)$

1-D DP

Maximal sum subarray (Problem 1.3.5)

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

$$A[-2, 1, -3, 4, -1, 2, 1, -5, 4] \Rightarrow [4, -1, 2, 1]$$

1-D DP

Maximal sum subarray (Problem 1.3.5)

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

$$A[-2, 1, -3, 4, -1, 2, 1, -5, 4] \Rightarrow [4, -1, 2, 1]$$

Trial and error.

- try subproblem MSS$[i]$: the sum of the MS (MS[i]) in $A[1 \cdots i]$
- goal: mss $=$ MSS$[n]$
- question: Is $a_i \in$ MS$[i]$?
- recurrence:

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

1-D DP

Solution.

- subproblem MSS$[i]$: the sum of the MS *ending with* a_i or 0
- goal: mss $= \max_{1 \le i \le n}$ MSS$[i]$

1-D DP

Solution.

- subproblem $\text{MSS}[i]$: the sum of the MS *ending with* a_i or 0
- goal: $\text{mss} = \max_{1 \le i \le n} \text{MSS}[i]$
- question: where does the $\text{MS}[i]$ start?
- recurrence:

$$\text{MSS}[i] = \max\{\text{MSS}[i-1] + a_i, 0\} \text{ (prove it!)}$$

1-D DP

Solution.

- subproblem MSS$[i]$: the sum of the MS *ending with* a_i or 0
- goal: mss $= \max_{1 \le i \le n}$ MSS$[i]$
- question: where does the MS$[i]$ start?
- recurrence:

$$\text{MSS}[i] = \max\{\text{MSS}[i-1] + a_i, 0\} \text{ (prove it!)}$$

- initialization: MSS$[0] = 0$

1-D DP

Code.

```
MSS[0] = 0
For i = 1 to n
  MSS[i] = max{MSS[i-1] + A[i], 0}
return max_{i = 1 to n} MSS[i]
```

1-D DP

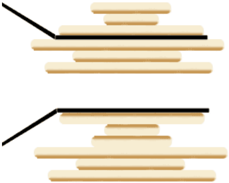
Code.

```
MSS[0] = 0
For i = 1 to n
  MSS[i] = max{MSS[i-1] + A[i], 0}
return max_{i = 1 to n} MSS[i]
```

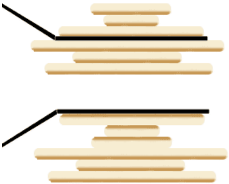
Simpler code.

```
mss = 0
MSS = 0
For i = 1 to n
  MSS = max{MSS + A[i], 0}
  mss = max{mss, MSS}
return mss
```

Pancake sorting (Problem 1.3.1)

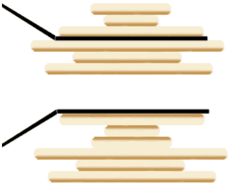


Pancake sorting (Problem 1.3.1)



How to bring the biggest pancake to the bottom?

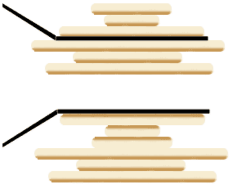
Pancake sorting (Problem 1.3.1)



How to bring the biggest pancake to the bottom?

$$T(n) = 2n - 3$$

Pancake sorting (Problem 1.3.1)



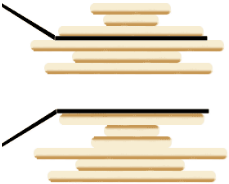
How to bring the biggest pancake to the bottom?

$$T(n) = 2n - 3$$

Reference

- $T(n) \leq \frac{5n+5}{3}$, 1979: "Sorting by Perfix Reversals"

Pancake sorting (Problem 1.3.1)



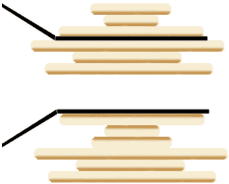
How to bring the biggest pancake to the bottom?

$$T(n) = 2n - 3$$

Reference

- $T(n) \leq \frac{5n+5}{3}$, 1979: "Sorting by Perfix Reversals" by Bill Gates *et al.*

Pancake sorting (Problem 1.3.1)



How to bring the biggest pancake to the bottom?

$$T(n) = 2n - 3$$

Reference

- $T(n) \leq \frac{5n+5}{3}$, 1979: "Sorting by Perfix Reversals" by Bill Gates *et al.*
- $T(n) \leq \frac{18n}{11}$, 2009

Big V's (Problem 1.3.8)

How many Big V's are there at most?

Big V's (Problem 1.3.8)

How many Big V's are there at most?

"Does A follow B?"

Big V's (Problem 1.3.8)

How many Big V's are there at most?

"Does A follow B?"

Don't forget to check it!

Repeated elements (Problem 2.12)

- $R[1 \ldots n]$
- check$(R[i], R[j])$
- $\# > \frac{n}{13}$
- $n \log n$

Repeated elements (Problem 2.12)

- $R[1 \dots n]$
- check$(R[i], R[j])$
- $\# > \frac{n}{13}$
- $n \log n$

We will talk about an $O(n \log k)$ algorithm and the lower bound.

Reference

"Finding Repeated Elements" by Misra & Gries, 1982

Bolts and nuts (Problem 2.10)

Bolts and nuts (Problem 2.10)

Using quicksort

Bolts and nuts (Problem 2.10)

Using quicksort

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

Bolts and nuts (Problem 2.10)

Using quicksort

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

Reference

$\Theta(n \log n)$ 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

Bolts and nuts (Problem 2.10)

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

Bolts and nuts (Problem 2.10)

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

Reducing it to the sorting problem.

Bolts and nuts (Problem 2.10)

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

Reducing it to the sorting problem.

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

K-sorted (Problem 2.9)

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

K-sorted (Problem 2.9)

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

1-sorted?

K-sorted (Problem 2.9)

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

1-sorted?
2-sorted?

K-sorted (Problem 2.9)

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

1-sorted?
2-sorted?
n-sorted?

K-sorted (Problem 2.9)

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

1-sorted?
2-sorted?
n-sorted?

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

K-sorted (Problem 2.9)

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

1-sorted?
2-sorted?
n-sorted?

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

Quicksort stops after the $\log k$ recursions.

K-sorted (Problem 2.9)

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

1-sorted?
2-sorted?
n-sorted?

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

Quicksort stops after the $\log k$ recursions.

$$O(n \log k)$$

K-sorted (Problem 2.9)

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

K-sorted (Problem 2.9)

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

$$L = \frac{n!}{\left(\left(\frac{n}{k}\right)!\right)^k}$$

K-sorted (Problem 2.9)

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

$$L = \frac{n!}{\left((\frac{n}{k})!\right)^k}$$

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

Dutch national flag problem (Problem 2.4)

The Dutch national flag

Edsger W. Dijkstra

Red balls *before* White balls *before* Blue balls

Dutch national flag problem (Problem 2.4)

The Dutch national flag

Edsger W. Dijkstra

Red balls *before* White balls *before* Blue balls

$\textsc{Color}(i) \quad \textsc{Swap}(i, j)$

Dutch national flag problem (Problem 2.4)

Loop invariant:

Dutch national flag problem (Problem 2.4)

Loop invariant:

$$r = 0;\ w = 0;\ b = n$$

Dutch national flag problem (Problem 2.4)

Loop invariant:

$$r = 0;\ w = 0;\ b = n$$

Red: $\textsc{Swap}(r, w);\ r = r + 1;\ w = w + 1;$

Dutch national flag problem (Problem 2.4)

Loop invariant:

$$r = 0;\ w = 0;\ b = n$$

Red: $\textsc{Swap}(r, w);\ r = r + 1;\ w = w + 1;$

White: $w = w + 1;$

Dutch national flag problem (Problem 2.4)

Loop invariant:

$$r = 0;\ w = 0;\ b = n$$

$$\begin{aligned}
\text{Red:}\quad & \textsc{Swap}(r, w);\ r = r + 1;\ w = w + 1; \\
\text{White:}\quad & w = w + 1; \\
\text{Blue:}\quad & \textsc{Swap}(b - 1, w);\ b = b - 1;
\end{aligned}$$


THANK YOU