

ACM Problem 11645 - Bits

Ausgewählte Probleme aus dem
ACM Programming Contest

Kevin Kratzer
Wintersemester 09/10

ACM Problem 11645 - Bits

- Problemstellung
- Lösungsidee
- Herleitung der Funktion
- Algorithmus
- Implementierung
- Theoretische Laufzeit

ACM Problem 11645 - Bits

- **Problemstellung**
- Lösungsidee
- Herleitung der Funktion
- Algorithmus
- Implementierung
- Theoretische Laufzeit

Problemstellung

- Ein Bit ist eine binäre Ziffer (0 oder 1)
- Jede Dezimalzahl kann binär als Serie von Bits dargestellt werden.

Dezimal	Binär
12	1100
15	1111
27	11011

Problemstellung

- Zwei benachbarte Bits mit dem Wert 1 nennt man "adjacent bit".

Dezimal	Binär
12	1100
15	1111
27	11011

Problemstellung

- Für jede gegebene Zahl N sollen alle vorkommenden adjacent bits von $0..N$ berechnet werden

Dezimal	Binär
0	000
1	001
2	010
3	0 11
4	100

Problemstellung

- Zeitlimit 1.0 Sekunden
- Multiple Eingabe

Eingabe	Ausgabe
0	Case 1: 0
6	Case 2: 2
15	Case 3: 12
20	Case 4: 13
21	Case 5: 13
22	Case 6: 14
-1	

ACM Problem 11645 - Bits

- Problemstellung
- **Lösungsidee**
- Herleitung der Funktion
- Algorithmus
- Implementierung
- Theoretische Laufzeit

Lösungsidee

- Herleitung einer Funktion für die Berechnung der Anzahl der adjacent bits in abhängigkeit der Bitanzahl
- Behandlung der Sonderfälle durch aufteilung des Problems in kleine Teilprobleme
 - Dynamische Programmierung
- Rekursion

ACM Problem 11645 - Bits

- Problemstellung
- Lösungsidee
- **Herleitung der Funktion**
- Algorithmus
- Implementierung
- Theoretische Laufzeit

ACM Problem 11645 – Bits

Herleitung der Funktion

Herleitung der Funktion

$$n(0) = 0$$

$$n(1) = 0$$

$$n(k) = 2 * n(k-1) + 2^{(k-2)}$$

ACM Problem 11645 - Bits

- Problemstellung
- Lösungsidee
- Herleitung der Funktion
- **Algorithmus**
- Implementierung
- Theoretische Laufzeit

Algorithmus

- Bit Länge der Zahl (N) : k
- Distanz ($2^k - 1 - N$): d

Algorithmus

- Bit Länge der Zahl (N) : k
- Distanz ($2^k - 1 - N$): d
- Wenn $k < 2$
→ $n(0)$

Algorithmus

- Bit Länge der Zahl (N) : k
- Distanz ($2^k - 1 - N$): d
- Wenn $k < 2$
 - $n(0)$
- Wenn $d = 0$
 - $n(k)$

Algorithmus

- Bit Länge der Zahl (N) : k
- Distanz ($2^k - 1 - N$): d
- Wenn $k < 2$
 - $n(0)$
- Wenn $d = 0$
 - $n(k)$
- Wenn $d = 2^{(k-2)}$
 - $n(k-1) + n(k-2)$

Algorithmus

- Bit Länge der Zahl (N) : k
- Distanz ($2^k - 1 - N$): d
- Sonst:
 - Wenn vorderstes Bit = 1: addiere $n(k-1)$
 - Wenn $d \leq 2^{(k-2)}$: addiere $2^{(k-2)} - d$
 - Streiche vorderstes Bit

ACM Problem 11645 - Bits

- Problemstellung
- Lösungsidee
- Herleitung der Funktion
- Algorithmus
- **Implementierung**
- Theoretische Laufzeit

ACM Problem 11645 - Bits

- Problemstellung
- Lösungsidee
- Herleitung der Funktion
- Algorithmus
- Implementierung
- **Theoretische Laufzeit**

Theoretische Laufzeit

- Ergebnisse der Formel einmal vorberechnet
- Die Zahl muss höchstens $(k - 1)$ mal verkürzt werden

→ Laufzeit linear: $O(k)$

ACM Problem 11645 - Bits

Vielen Dank für die Aufmerksamkeit

Fragen?