

CSE 20

DISCRETE MATH

WINTER 2016

<http://cseweb.ucsd.edu/classes/wi16/cse20-ab/>

Today's learning goals

- Define the decimal, binary, hexadecimal, and octal expansions of a positive integer.
- Describe and use algorithms for integer operations based on their expansions
- Define and use the DIV and MOD operators.

About you

Remote frequency: CA

To change your remote frequency

1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

How many people in this class have you met so far?

- A. None.
- B. Less than 5.
- C. 5-10.
- D. 10-15.
- E. More than 15.

Algorithms!

From last time

An **algorithm** is a finite sequence of precise instructions for performing a computation or for solving a problem.

... arithmetic

... optimization

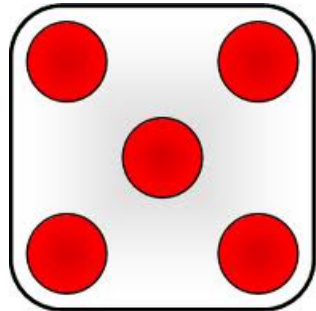


Representation



$$\log_2(32)$$

Five



101



Integer representations

Different contexts call for different representations.



Base 10



Base 2

Bases

Are base 2 and base 10 the only possible ways to express positive integers?

- A. Yes.
- B. No: there's also base 8 (octal) and base 16 (hexadecimal), but that's it.
- C. No: there's a few other special possible bases.
- D. No: any integer greater than 1 can be a base.

Base expansion

For what positive integer k can we choose $a_0, a_1, a_2, a_3, \dots, a_k$ so that $a_k \neq 0$ and each a_i is 0 or 1 with

$$17 = a_k 2^k + a_{k-1} 2^{k-1} + \dots + a_1 2 + a_0$$

A. $k = 17$

B. $k = 2$

C. $k = 4$

D. $k = 5$

E. Multiple values of k would work.

Base expansion

For what positive integer k can we choose

$a_0, a_1, a_2, a_3, \dots, a_k$ so that $a_k \neq 0$ and each a_i is 0 or 1 with

$$17 = a_k 2^k + a_{k-1} 2^{k-1} + \dots + a_1 2 + a_0$$

Writing 17 as a sum of powers of 2!

Base expansion

Find positive integer k and coefficients $a_0, a_1, a_2, a_3, \dots, a_k$ so that

$$\begin{aligned}17 &= a_k 2^k + a_{k-1} 2^{k-1} + \dots + a_1 2 + a_0 \\ &= 16 + 1 \\ &= 1 \cdot 2^4 + 1\end{aligned}$$

So $k=4$ and $a_4=1, a_3=0, a_2=0, a_1=0, a_0=1$.

Base expansion

Rosen p. 246

For base b (**integer greater than 1**) and positive integer n there is unique choice of

- k , a nonnegative integer
- $a_0, a_1, a_2, a_3, \dots, a_k$ integers between 0 and $b-1$, where
- $a_k \neq 0$

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0$$

Base expansion

Rosen p. 246

Notation: for positive integer n

Write

$$(a_k a_{k-1} \dots a_1 a_0)_b$$

when

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0$$

Base b expansion of n

Common bases

Rosen p. 246

- **Decimal expansion**

base 10

(17)₁₀

- **Binary expansion**

base 2

(10001)₂

- **Octal expansion**

base 8

()₈

- **Hexadecimal expansion**

base 16

()₁₆

Common bases

Rosen p. 246

- **Decimal expansion**

base 10

(17)₁₀

- **Binary expansion**

base 2

(10001)₂

- **Octal expansion**

base 8

(21)₈

- **Hexadecimal expansion**

base 16

(11)₁₆

Common bases

- **Decimal expansion**
- **Binary expansion**
- **Octal expansion**
- **Hexadecimal expansion**

base 10

base 2

base 8

base 16

Rosen p. 246

(17)₁₀

(10001)₂

(21)₈

(11)₁₆

What's different?

Base expansion

In what base could this expansion be
 $(1401)_?$

- A. Binary (base 2)
- B. Octal (base 8)
- C. Decimal (base 10)
- D. Hexadecimal (base 16)
- E. More than one of the above

Base expansion

In what base could this expansion be
 $(1401)_?$

A. Binary (base 2)

B. Octal (base 8)

$$\text{Value: } 1 \cdot 8^3 + 4 \cdot 8^2 + 1 = 768$$

C. Decimal (base 10)

$$\text{Value: } 1 \cdot 10^3 + 4 \cdot 10^2 + 1 = 1401$$

D. Hexadecimal (base 16)

$$\text{Value: } 1 \cdot 16^3 + 4 \cdot 16^2 + 1 = 5121$$

E. More than one of the above

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

- Pseudocode.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

Find k

Work down to find a_k , then a_{k-1} , etc.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

Find k

Work down to find a_k , then a_{k-1} , etc.

What's the length of the base b expansion of n ?

Start with related question:

What's the biggest number that can be represented with k digits?

Bounds

Base 2

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

Base 3

0	0	102	11	211	22
1	1	110	12	212	23
2	2	111	13	220	24
10	3	112	14	221	25
11	4	120	15	222	26
12	5	121	16	1000	27
20	6	122	17	1001	28
21	7	200	18	1002	29
22	8	201	19	1010	30
100	9	202	20	1011	31
101	10	210	21	1012	32

Bounds

Base 2

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

What's the biggest number whose base b expansion has k digits, i.e.

$$(a_{k-1} \dots a_1 a_0)_b?$$

- A. b^k
- B. b^{k-1}
- C. $b^{k-1}-1$
- D. $b^{k-1}-1$
- E. None of the above.

Bounds

Base 2

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

What's the biggest number whose base b expansion has k digits, i.e.

$$(a_{k-1} \dots a_1 a_0)_b?$$

- A. b^k
- B. b^{k-1}
- C. $b^{k-1}-1$
- D. $b^{k-1}-1$
- E. None of the above.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b

Output k , coefficients in expansion

- English description.

Find k by computing successive powers of b until find smallest k such that

$$b^{k-1} \leq n < b^k$$

For each value of i from 1 to k

Set a_{k-i} to be the largest number between 0 and $b-1$ for which $a_{k-i} b^{k-i} < n$.

Update current value remaining $n := n - a_{k-i} b^{k-i}$

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.
Find k by computing successive powers of b until find smallest k such that

$$b^{k-1} \leq n < b^k$$

For each value of i from 1 to k

Set a_{k-i} to be the largest number between 0 and $b-1$ for which $a_{k-i} b^{k-i} < n$.

Update current value remaining $n := n - a_{k-i} b^{k-i}$

What's the the base 3 expansion of 17?

- A. $(10001)_3$
- B. $(210)_3$
- C. $(111)_3$
- D. $(222)_3$
- E. None of the above.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n,b **Output** k, coefficients in expansion

- English description.

Find k by computing successive powers of b until find smallest k such that

$$b^{k-1} \leq n < b^k$$

For each value of i from 1 to k

Set a_{k-i} to be the largest number between 0 and b-1 for which $a_{k-i} b^{k-i} < n$.

Update current value remaining $n := n - a_{k-i} b^{k-i}$

Definite? Finite? Correct?

Challenge: translate to pseudocode!

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

Idea: Find smallest digit first, then next smallest, etc.

.... **but how?**

- Pseudocode.

Bases and Divisibility

Rosen p. 237-239

When $k > 0$

$$\begin{aligned} n &= a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0 \\ &= b (a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1) + a_0 \end{aligned}$$

Quotient when
divide n by b

Remainder
when divide n
by b

Bases and Divisibility

Rosen p. 237-239

When $k > 0$

$$\begin{aligned}n &= a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0 \\ &= b (a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1) + a_0\end{aligned}$$

Quotient when
divide n by b

Remainder
when divide n
by b

Theorem: For a an integer and d a positive integer, there are unique integers q and r with $0 \leq r < d$ and $a = qd + r$.

Notation: $q = a \operatorname{div} d$

$r = a \operatorname{mod} d$

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

 Compute $n \bmod b$ to obtain a_0 .

 Update value $n := n \operatorname{div} b$ of integer whose expansion we need.

 Repeat.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- English description.

Compute $n \bmod b$ to obtain a_0 .

Update value $n := n \operatorname{div} b$ of integer whose expansion we need.

Repeat.

How is the base b expansion of n related to the base b expansion of $n \operatorname{div} b$?

- A. They're the same.
- B. They're unrelated.
- C. The expansion of n has an extra 0 at the end.
- D. The expansion of n has an extra 0 at the front.

Almost but not quite!
See next slide...

Fixing question response C:

$$\begin{aligned}n &= a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0 \\ &= b (a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1) + a_0\end{aligned}$$

So base b expansion of n is

$$(a_k a_{k-1} \dots a_1 a_0)_b$$

and $n \text{ div } b$ is $a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1$, which has the expansion

$$(a_k \dots a_1)_b$$

So: base b expansion of n is the result of taking the base b expansion of $n \text{ div } b$, shifting one column to the left and *adding a coefficient* appended to the end.

Algorithm: constructing base b expansion *Rosen p. 249*

Input n, b **Output** k , coefficients in expansion

- Pseudocode.

procedure *base b expansion*(n, b : pos ints with $b > 1$)

1. $q := n$
2. $k := 0$
3. **while** $q \neq 0$
4. $a_k := q \bmod b$
5. $q := q \text{ div } b$
6. $k := k + 1$
7. **return** $(a_{k-1}, \dots, a_1, a_0)$

Algorithm: constructing base b expansion *Rosen p. 249*

procedure *base b expansion*(n, b : pos ints with $b > 1$)

1. $q := n$
2. $k := 0$
3. **while** $q \neq 0$
4. $a_k := q \bmod b$
5. $q := q \operatorname{div} b$
6. $k := k + 1$
7. **return** $(a_{k-1}, \dots, a_1, a_0)$

n	b	q	k	a_k
17	3	17	0	$17 \bmod 3 = 2$
		$17 \operatorname{div} 3 = 5$	1	$5 \bmod 3 = 2$
		$5 \operatorname{div} 3 = 1$	2	$1 \bmod 3 = 1$
		$1 \operatorname{div} 3 = 0$	3	return!

Definite? Finite? Correct?

Arithmetic + Representations

Rosen p. 251

What is the sum of $(A)_{16}$ and $(12)_{16}$?

- A. $(12A)_{16}$
- B. $(1A)_{16}$
- C. $(22)_{16}$
- D. $(16)_{16}$
- E. $(1C)_{16}$

Hexadecimal digits

0	8
1	9
2	A
3	B
4	C
5	D
6	E
7	F

Arithmetic + Representations

Rosen p. 251

What is the product of $(A)_{16}$ and $(12)_{16}$?

- A. $(B4)_{16}$
- B. $(78)_{16}$
- C. $(12A)_{16}$
- D. $(120)_{16}$
- E. $(A12)_{16}$

Hexadecimal digits

0	8
1	9
2	A
3	B
4	C
5	D
6	E
7	F

Arithmetic + Representations

Rosen p. 251

procedure *mystery1*(a, b : pos ints with binary expansions $(a_{n-1}, \dots, a_0), (b_{n-1}, \dots, b_0)$)

1. $c := 0$
2. **for** $j := 0$ **to** $n - 1$
3. $d := (a_j + b_j + c) \text{ div } 2$
4. $s_j := a_j + b_j + c - 2d$
5. $c := d$
6. $s_n := c$
7. **return** (s_n, \dots, a_1, a_0)

Trace this procedure on inputs

$$a = 5 = (101)_2, b = 4 = (100)_2$$

What is the return value?

Arithmetic + Representations

Rosen p. 251

procedure *mystery1*(a, b : pos ints with binary expansions $(a_{n-1}, \dots, a_0), (b_{n-1}, \dots, b_0)$)

1. $c := 0$
2. **for** $j := 0$ **to** $n - 1$
3. $d := (a_j + b_j + c) \text{ div } 2$
4. $s_j := a_j + b_j + c - 2d$
5. $c := d$
6. $s_n := c$
7. **return** (s_n, \dots, a_1, a_0)

What's a higher-level description of the function computed by this procedure?

- A. It adds up the bits of the binary expansions and divides each by 2.
- B. It returns the binary expansion of a number that is the sum of the two input integers, divided by 2.
- C. It returns the binary expansion of the sum of the two input integers.
- D. None of the above.

Arithmetic + Representations

Rosen p. 251

procedure *mystery1*(a, b : pos ints with binary expansions $(a_{n-1}, \dots, a_0), (b_{n-1}, \dots, b_0)$)

1. $c := 0$
2. **for** $j := 0$ **to** $n - 1$
3. $d := (a_j + b_j + c) \text{ div } 2$
4. $s_j := a_j + b_j + c - 2d$
5. $c := d$
6. $s_n := c$
7. **return** (s_n, \dots, a_1, a_0)

How would we change this procedure to compute the base b expansion of the sum $a + b$, given as input the base b expansions of a and b ?

Arithmetic + Representations

Rosen p. 251

procedure *mystery1*(a, b : pos ints with binary expansions $(a_{n-1}, \dots, a_0), (b_{n-1}, \dots, b_0)$)

1. $c := 0$
2. **for** $j := 0$ **to** $n - 1$
3. $d := (a_j + b_j + c) \text{ div } 2$
4. $s_j := a_j + b_j + c - 2d$
5. $c := d$
6. $s_n := c$
7. **return** (s_n, \dots, a_1, a_0)

On Tuesday, we'll see how the binary case works well with logic circuits and Boolean algebra...

Reminders

- Homework 1 due tomorrow
 - Set up course tools
 - Pseudocode and algorithms
- Office hours