実数，文字

実数の表現

計算機の中では常に有限のビットで数値を表す

- n ビットでは 2^n 個の状態しか区別できない
- 実数も，有限の精度，有限の範囲でしか表現できない

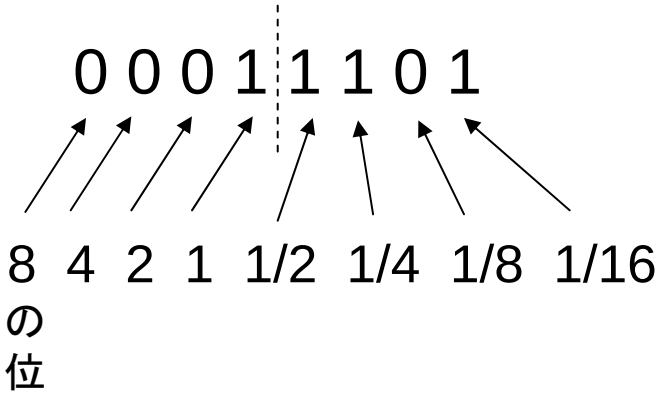
精度や範囲が有限であることを理解していないとひどい目に合う

例: 以下のコードは，0.1, 0.2, ..., 1.0 の10個の数の指数関数を表示して…くれない

```
float x;
for (x = 0.1; x <= 1.0; x = x + 0.1) {
    printf(”exp(%f) = %f¥n”, x, exp(x));
}
```

固定小数点数

あらかじめ決まった箇所に小数点があると考える



- 科学技術計算に必要な数値範囲を十分に表せない
- 小さな数は有効桁数が少なく，大きな数は（しばしば不必要なくらい）有効桁数が多い

浮動小数点数

いわゆる科学的記数法（指数表記）

- 6.02×10^{23}
- 1.602×10^{-19}

  仮数部　　指数部

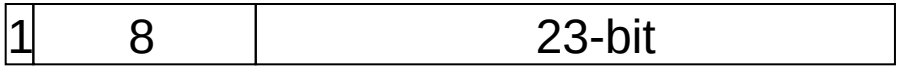
- 仮数部は1.0以上10.0未満にする（正規化）
- 仮数部の桁数がいわゆる有効数字

これの2進数版が浮動小数点数

- 仮数部も指数も有限ビットの2進数で表す
- 指数部の底は 10 ではなく 2

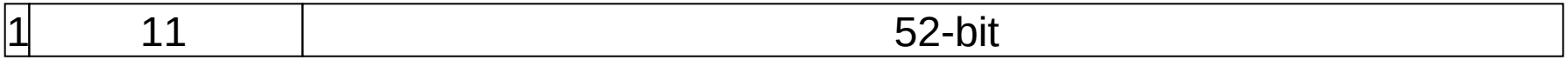
IEEE 754 浮動小数点数

単精度 (C言語の float)



符号　指数部　　仮数部

倍精度 (C言語の double)



符号　指数部　　仮数部

$$-1.01101101 \times 2^{1100110}$$

符号:
0: 正
1: 負

仮数部

指数部

IEEE 754: 仮数部

正規化: $1.xxxxxxx_{(2)}$ の形になるようにする（1以上，2未満）

最小値: $1.0_{(2)} = 1.0_{(10)}$

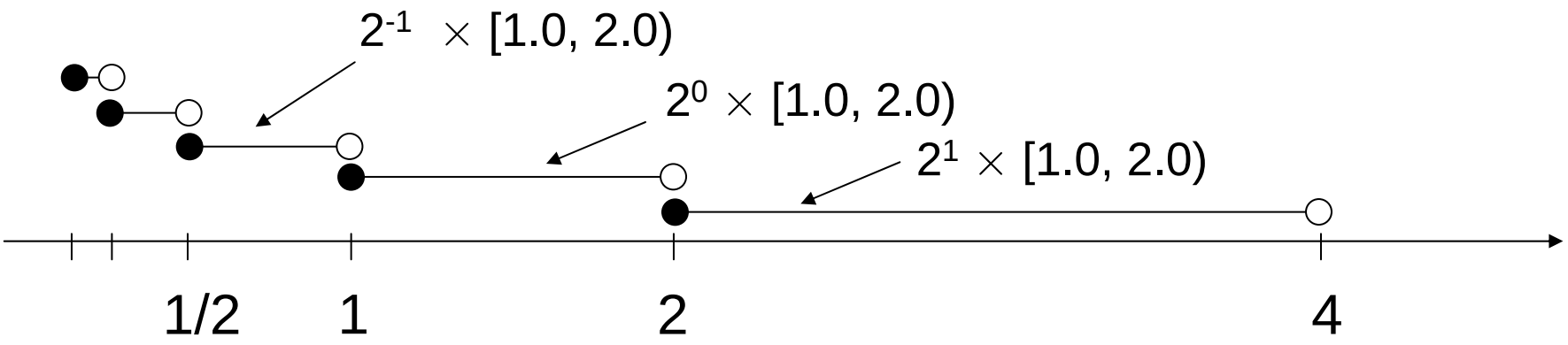
最大値: $1.1111\cdots_{(2)} = 1.99999\cdots_{(10)}$

1の位は正規化の結果つねに 1 なので，仮数部には含めない
（economized form, けち表現）

例: 仮数部のビット列が 0110100 … 00

$\rightarrow 1.01101_{(2)} = (1 + 1/4 + 1/8 + 1/32)_{(10)}$

$= 1.40625_{(10)}$



各区間は，2^{23} 等分（単精度）または 2^{52} 等分（倍精度）

IEEE 754: 指数部

- 指数部が n ビット長のとき，指数に $2^{n-1} - 1$ を足した値を符号なし整数として表し，指数部に収める（biased form, ゲタばき表現）
- ただし「全ビット0」と「全ビット1」は特別扱い

  - 単精度: 指数部は8ビット長なので，バイアス値は $2^7 - 1 = 127$．よって -127 ～ 128 にバイアス値を加えて符号なし数 0 ～ 255 にする．このうち全ビット0（= 0）と全ビット1（= 255）は除くので，結局，指数として可能なのは -126 ～ 127
  - 倍精度: 指数部は11ビット長なので，バイアス値は $2^{10} - 1 = 1023$．同様に考えて，指数として可能なのは -1022 ～ 1023

例: 単精度浮動小数点数の指数部のビット列が 10110101
→ $10110101_{(2)} = (128+32+16+4+1)_{(10)} = 181_{(10)}$
→ バイアス値を引いて，181 – 127 = 54 が指数の値

IEEE 754: 特殊な数

指数部が全ビット 0 のとき:

- 指数を最小値（単精度で -126, 倍精度で -1022）とし，仮数は1の位を 0 として仮数部から組み立てる（非正規化数，有効桁数が落ちることに注意）．特に，全ビット0は数0になる

指数部が全ビット 1 のとき:

- 仮数部が全ビット0: 無限大 (Inf)
  - 符号ビットが0, 1 のとき，それぞれ $+\infty$, $-\infty$
- その他の仮数部: Not a Number (NaN)
  - 負数の平方根や，0 / 0 など

例: 単精度で指数部 0000 0000，仮数部 0110100 … 00

→ 0.40625×2^{-126}

IEEE 754: 丸め

与えられた精度内で表現できない数は「近い」数に丸める.
IEEE 754 は 4 つの丸めモードを定義:

- Round to nearest – even（通常はこれを用いる）
  - 表現可能な値のうち最も近いものに丸める
  - 最も近いものが2つある場合，仮数部のLSBが0のものを選ぶ
- Towards zero
- Towards positive infinity
- Towards negative infinity

例

10進数 3.25 → 単精度浮動小数点数

- 3.25 は 2^1 以上，2^2 未満．よって指数は 1
- バイアス値 127 を履かせて，指数部は $128_{(10)}$ = 1000 0000
- $3.25 = 1.625 \times 2^1$
- $1.625 = 1 + 1/2 + 1/8 = 1.101000 \cdots 00_{(2)}$
- よって仮数部は 101000 … 000
- 正なので符号ビットは 0
- まとめると，0 1000 0000 101000 … 000

例

10進数 0.1 → 単精度浮動小数点数

- 0.1 は 2^{-4} 以上，2^{-3} 未満．よって指数は -4
- バイアス値 127 を履かせて，指数部は $123_{(10)}$ = 0111 1011
- 0.1 = 1.6× 2^{-4}
- 1.6 = 1 + 1/2 + 1/16 + ... と考えてみてもすぐに分解できそうにないので，真面目に 16/10 または 8/5 などを2進数で筆算してみるとよい．すると $1.10011001100\cdots_{(2)}$ と循環することがわかる．
- 仮数部は 23 ビット長なので，
  1001 1001 1001 1001 1001 100
  までは入り，以降の 1100... が丸められる．round to nearest により切り上げ．
- 正なので符号ビットは 0
- まとめると，0 0111 1011 1001 1001 1001 1001 1001 101

加減算

1. 桁あわせ: 絶対値の大きな方の指数部に合わせる
2. 仮数部の加減算
3. 正規化
4. 丸め（必要に応じて再度正規化）
5. 符号をセット

$$
\begin{aligned}
& 1.5 \times 2^8 + 1.75 \times 2^6 \\
= & (1.5 + 0.4375) \times 2^8 \\
= & 1.9375 \times 2^8
\end{aligned}
$$

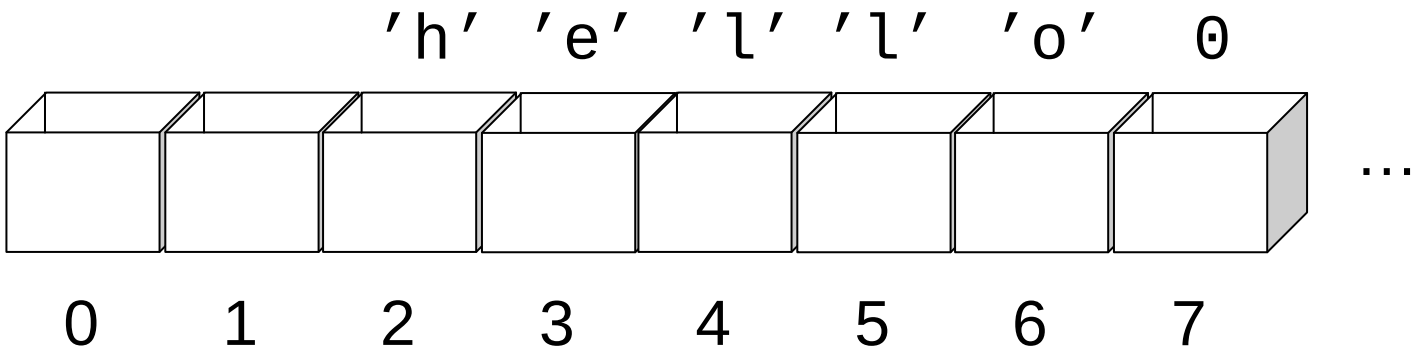
乗除算

1. 指数部の加減算
2. 仮数部の乗除算
3. 正規化
4. 丸め（必要に応じて再度正規化）
5. 符号をセット

$$
\begin{aligned}
&(1.5 \times 2^{8}) \times (1.75 \times 2^{6}) \\
&= (1.5 \times 1.75) \times 2^{8+6} \\
&= 2.625 \times 2^{14} \\
&= 1.3125 \times 2^{15}
\end{aligned}
$$

文字と文字列

```
char text[] = “hello”;
```



「文字」も数字の組み合わせで表現

例) ASCIIコード
1 文字を 1 バイト (ただし下位 7 ビットのみを使用) で表す.

Cでは, 文字は char 型で, 文字列は char の配列型で表す.
文字列の終端を示すために 0('¥0')を用いる.

例: ASCIIコード表

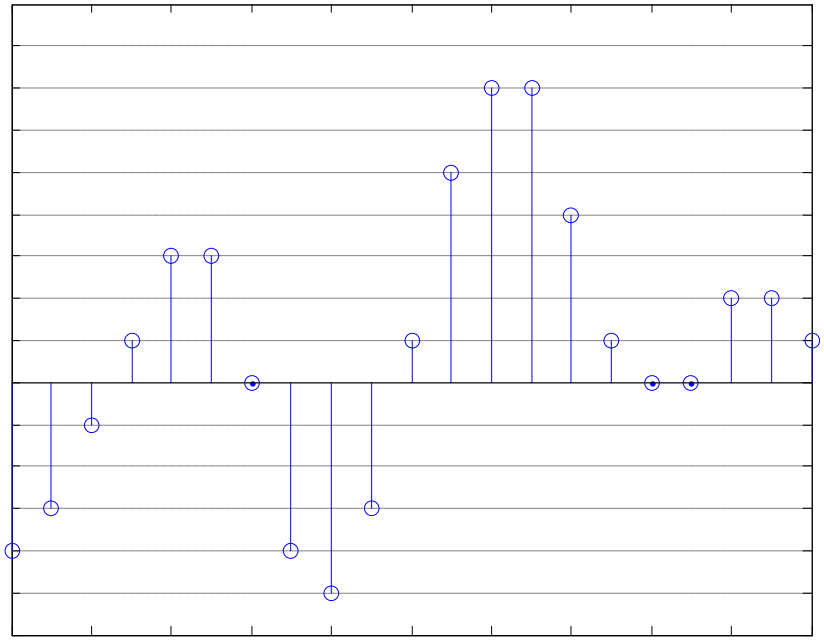
下位4ビット＼上位3ビット	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAC	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF/NL	SUB	×	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

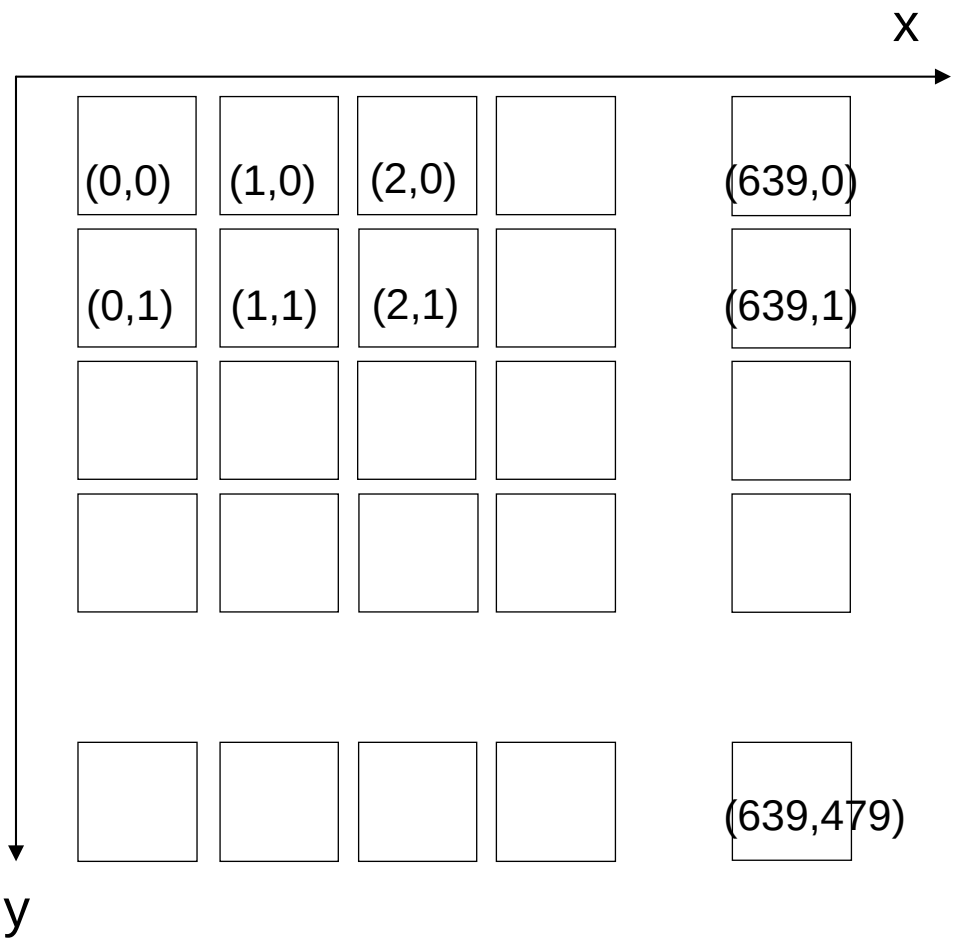
http://itpro.nikkeibp.co.jp/article/COLUMN/20060929/249401/

その他のデータ

音声も画像も数値の集まりとして表現

すべてのものはビットの集まりである





練習問題

IEEE 754 単精度浮動小数点数フォーマットにおいて，仮数部のみ 5 ビット長に変更したものを考える．丸めは 0 方向へ行うものとする．

1) 以下の10進数をこのフォーマットの浮動小数点数で表せ
   (a) x = 20.1　(b) y = 1.1

2) 上の数 x, y に対して以下の浮動小数点数演算を行い，その過程を示せ
   (a) x + y　　(b) xy

練習問題 解答例

1) (a) 20.1 は 2^4 以上 2^5 未満. よって指数は 4
   - バイアス値 127 を履かせて，指数部は $131_{(10)}$ = 1000 0011
   - 20.1 = 1.25625$\times$ 2^4
   - 1.25625 = 1 + 1/4 + 1/256 + …= 1.01000 …$_{(2)}$
   - 「0 方向への丸め」なので，仮数部5ビットは 01000
   - 符号ビット 0 をつけて，答えは 0 1000 0011 01000

   (b) 1.1 は 2^0 以上 2^1 未満. よって指数は 0
   - バイアス値 127 を履かせて，指数部は$127_{(10)}$ = 0111 1111
   - 1.1 = 1 + 1/16 + 1/32 + … = 1.00011 …$_{(2)}$
   - 符号ビット 0 をつけて，答えは 0 0111 1111 00011

2) (a) $1.01000_{(2)} \times 2^4 + 1.00011_{(2)} \times 2^0$
   $= (1.01000 + 0.000100011) \times 2^4 \fallingdotseq 1.01010 \times 2^4$

   (b) $1.01000_{(2)} \times 2^4 \times 1.00011_{(2)} \times 2^0$
   $= 1.0101111 \times 2^4 \fallingdotseq 1.01011 \times 2^4$