“Finite” Number Representations

- Multiple meanings for “finite”
- Bits = *binary integer* = *(0, 1)* = *(↑, ↓)*
- All computer numbers ultimately binary
- Integers*:* *N* bits ⇒ 2^N max
- ⇒ Limited range $[0, 2^{N-1}]$ (1 sign, 32 b ⇒ 10^9)
- Binary: 110010010101:   OK if computer
- People: *octal, decimal, hexadecimal* ()
- Decimal: nice but reduces precision, final output

Finite Number Representations (2)

- *"Word length"* = # bits for stored number
- 1 byte = 1 B = 8 bits ()
- Careful: 1 K = 1 KB = 2^{10} B =1024 B
- 1 byte ≈ memory for 1 character ("a")
- 1 typed page ≈ 3 KB
- 1st PCs: 8-b words (2^7= 128)
- *"Overflow"*: too large a number; *"Underflow"*
- Now: 32, 64 bit PC, fastest processing
- $2^{31} \approx 2 \times 10^9$: OK for banks, signals, not science

Fixed-point Numbers (ints)

$$I_{fix} = \pm \left(\alpha_n 2^n + \alpha_{n-1} 2^{n-1} + \cdots + \alpha_0 2^0 + \cdots + \alpha_{-m} 2^{-m}\right) \qquad (1)$$

- *1* bit: sign, *N-1* bits: α_i
- *N*, *m*, *n* values: machine-dependent
- 32-bit machine $\rightarrow$ 4B for *int*
- *Good*: absolute error $\equiv 2^{-(m+1)}$ (left off)
- *Bad*: large relative error for small #
- *Bad*: small range:

  $-2147483648 \leq \text{int} \leq 2147483647$

IEEE Number Types

Name	Type	Bits	Range
boolean	logical	1	*true* or *false*
char	string	16	'\u0000' ⇔ '\uFFFF'
byte	integer	8	-128 ⇔ +127
short	integer	16	-32,768 ⇔ +32,767
int	integer	32	-2,147,483,648 ⇔ +2,147,483,647
long	integer	64	
-9,223,372,036,854,775,808 ⇔ +9,223,372,036,854,775,807			
float	floating point	32	1.401298×10^{-45} ⇔ $3.402923 \times 10^{+38}$
double	floating point	64	
$4.94065645841246544 \times 10^{-324}$ ⇔ $1.7976931348623157 \times 10^{+308}$			

IEEE Floating Point

- **Binary version of scientific notation**
- ***c*** **≈ +2.997924 x 10^8 m/sec**
- **2.997924 = mantissa,**
  7 significant figures
- **+8 = exponent**
- **. = *decimal* point (base 10)**
- **Storage: (sign) (exponent) (mantissa)**

IEEE Standard Float (how)*

$$x_{\text{Float}} = (-1)^s \times 1.f \times 2^{\text{e - bias}} \qquad (1)$$

- Sign *s* = single bit = 0 (+), 1 (-)
- *f* = fractional part after *binary* point
  - assume 1st bit = 1 (phantom)
  - maintains same relative precision
- e = stored exponent always > 0
- bias: fixed, e < bias $\Rightarrow$ p = true exp < 0
- *Normal* numbers: 0 < e < 255
- *Subnormal* numbers: e=0, e=255
  - special cases & numbers (table)

IEEE Special Cases

Number Name	*Values of s, e & f*	*Value of Single*
Normal	$0 < e < 255$	$(-1)^s \times 2^{e-127} \times 1.f$
Subnormal	$e = 0, \;\; f \neq 0$	$(-1)^s \times 2^{-126} \times 0.f$
Signed Zero	$e = 0, \;\; f = 0$	$(-1)^s \times 0.0$
$+\infty$ ($\neq$ math)	$s = 0, \;\; e = 255, \;\; f = 0$	+INF
$-\infty$ ($\neq$ math)	$s = 1, \;\; e = 255, \;\; f = 0$	-INF
Not a Number	$s = u, \;\; e = 255, \;\; f \neq 0$	NaN

Implementation: IEEE Single (float)*

Position	s	e	f
32 Bit word	31	30 23	22 0

- **Conversion of Exponent *e***
  - biased exponent *e*: 8 bits

$$(-1)^s \times 1.f \times 2^{e-127} \quad (1)$$

  - normal: $0 < e < 255$
  - $\Rightarrow \quad 1 \leq e \leq 254$
  - bias = $127_{10} \quad \Rightarrow p = e_{10} - 127$
  - $-126 \leq p \leq 127$ (see limits)
- **Specials**
  - $e = f = 0: \quad \pm 0$

$$(-1)^s \times 0.f \times 2^{e-126} \quad (2)$$

  - $e = 0, \; f \neq 0$: mantissa = $0.f$

E.G.: Largest, Normal, 32-bit Float*

$$
\begin{aligned}
e_{max} \text{ (normal)} &= 254 \quad \Rightarrow \quad p = e - 127 = 127 && (1)\\
s &= 0 && (2)\\
f\text{max} &= 1.1111\ 1111\ 1111\ 1111\ 1111\ 111 && (3)\\
&= 1 + 0.5 + 0.25 + \ldots \simeq 2 && (4)\\
\Rightarrow & \quad (-1)^s \times 1.f \times 2^{p=e-127} \simeq 2 \times 2^{127} && (5)\\
&\simeq 3.4 \times 10^{38} && (6)
\end{aligned}
$$

