Rappresentazione Binaria dei Numeri Decimali e Virgola Mobile

1. I fondamenti della conversione tra sistema decimale e binario
2. Come convertire la parte intera e frazionaria dei numeri
3. Il problema dei numeri periodici in binario
4. La rappresentazione in virgola mobile secondo lo standard IEEE 754
5. Le cause e le conseguenze degli errori di arrotondamento
6. Esercizi pratici per consolidare la comprensione

La lezione è strutturata in modo progressivo, partendo dai concetti base fino ad arrivare a quelli più complessi, con esempi pratici per facilitare la comprensione. Puoi utilizzarla come materiale didattico, modificando le sezioni in base al livello di dettaglio che desideri raggiungere con i tuoi studenti.

# Obiettivi della lezione

 Comprendere come si rappresentano i numeri decimali in formato binario  Capire il concetto di numeri in virgola mobile e lo standard IEEE 754

 Esaminare vantaggi e limitazioni delle rappresentazioni binarie dei numeri reali

# Introduzione ai sistemi numerici

## Sistema decimale vs Sistema binario

Nel sistema decimale (base 10) usiamo le cifre da 0 a 9 e ogni posizione rappresenta una potenza di 10:  123,45 = 1×10² + 2×10¹ + 3×10⁰ + 4×10⁻¹ + 5×10⁻²

Nel sistema binario (base 2) usiamo solo 0 e 1, con ogni posizione che rappresenta una potenza di 2:

 101,11 = 1×2² + 0×2¹ + 1×2⁰ + 1×2⁻¹ + 1×2⁻²  In decimale: 4 + 0 + 1 + 0,5 + 0,25 = 5,75

# Conversione da decimale a binario

## Conversione della parte intera

Si divide ripetutamente il numero per 2, annotando i resti (dal basso verso l'alto): Esempio con 25:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2255 | ÷÷ | 22 | == | 1122 | ccoonn | rreessttoo | 11 |
| 1122 | ÷÷ | 22 | == | 66 | ccoonn | rreessttoo | 00 |
| 66 | ÷÷ | 22 | == | 33 | ccoonn | rreessttoo | 00 |
| 33 | ÷÷ | 22 | == | 11 | ccoonn | rreessttoo | 11 |
| 11 | ÷÷ | 22 | == | 00 | ccoonn | rreessttoo | 11 |

Quindi 25₁₀ = 11001₂

## Conversione della parte frazionaria

Si moltiplica ripetutamente per 2, annotando la parte intera (dall'alto verso il basso): Esempio con 0,625:

0,625 × 2 = 1,25 → prendi 1

0,25 × 2 = 0,5 → prendi 0

0,5 × 2 = 1,0 → prendi 1

Quindi 0,625₁₀ = 0,101₂

## Esempio completo

Convertire 25,625₁₀ in binario:

 Parte intera: 25₁₀ = 11001₂

 Parte frazionaria: 0,625₁₀ = 0,101₂

 Numero completo: 25,625₁₀ = 11001,101₂

# Numeri decimali periodici in binario

Non tutti i numeri con rappresentazione finita in decimale hanno rappresentazione finita in binario. Esempio: 0,1₁₀ in binario

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00,,11 | ×× | 22 | == | 00,,22 | →→ | pprreennddii | 00 |
| 00,,22 | ×× | 22 | == | 00,,44 | →→ | pprreennddii | 00 |
| 00,,44 | ×× | 22 | == | 00,,88 | →→ | pprreennddii | 00 |
| 00,,88 | ×× | 22 | == | 11,,66 | →→ | pprreennddii | 11 |
| 00,,66 | ×× | 22 | == | 11,,22 | →→ | pprreennddii | 11 |
| 00,,22 | ×× | 22 | == | 00,,44 | →→ | pprreennddii | 00 ((ee qquuii iinniizziiaa aa rriippeetteerrssii)) |

Quindi 0,1₁₀ = 0,00011001100110...₂ (periodico)

Questo è un concetto fondamentale per comprendere gli errori di arrotondamento nei computer.

# Rappresentazione in virgola mobile (Floating Point)

## Problemi della rappresentazione a virgola fissa

 Range limitato

Precisione uniforme non ottimale



## Concetto di virgola mobile

La rappresentazione in virgola mobile segue il formato:

numero = ±significando × base^esponente

Nel sistema decimale: 3,14159 × 10⁸ = 314.159.000 Nel sistema binario: 1,101 × 2³ = 1101 (13 in decimale)

## Lo standard IEEE 754

Lo standard più diffuso per la rappresentazione in virgola mobile nei computer.

### Formato a 32 bit (singola precisione):

 1 bit per il segno (0 positivo, 1 negativo)  8 bit per l'esponente (con bias di 127)

 23 bit per la mantissa (con 1 implicito)

### Formato a 64 bit (doppia precisione):

 1 bit per il segno

 11 bit per l'esponente (con bias di 1023)  52 bit per la mantissa (con 1 implicito)

## Normalizzazione

Il significando viene normalizzato in modo che abbia sempre un solo bit a sinistra del punto binario: 1,xxxxx × 2ᵉ

Il bit più significativo (1) è implicito nella rappresentazione IEEE 754.

## Esempi di rappresentazione IEEE 754

Rappresentazione di 10,5 in singola precisione:

* + 1. Converti in binario: 10,5₁₀ = 1010,1₂
    2. Normalizza: 1,0101 × 2³
    3. Segno: positivo → 0
    4. Esponente: 3 + 127 (bias) = 130 → 10000010
    5. Mantissa: 0101 seguito da zeri → 01010000000000000000000
    6. Rappresentazione finale: 01000001001010000000000000000000

## Casi speciali

 **Zero**: esponente = 0, mantissa = 0

 **Infinito**: esponente = tutti 1, mantissa = 0

 **NaN** (Not a Number): esponente = tutti 1, mantissa ≠ 0

 **Numeri denormalizzati**: per valori molto piccoli

# Errori di arrotondamento

## Cause degli errori

 Rappresentazione finita di numeri infiniti  Conversione tra basi numeriche

 Operazioni aritmetiche con precisione limitata

## Esempio pratico

0,1 + 0,2 ≠ 0,3 (esattamente) nei computer

Perché 0,1 e 0,2 hanno rappresentazioni periodiche in binario, che vengono troncate.

## Propagazione degli errori

Piccoli errori di arrotondamento possono accumularsi in calcoli complessi.

## Strategie per gestire gli errori

 Confronto con tolleranza invece dell'uguaglianza esatta

 Utilizzo di tipi di dati per numeri decimali esatti (es. fixed-point, decimal)  Algoritmi numericamente stabili

# Esercizi pratici

## Conversioni

* + 1. Convertire 42,75₁₀ in binario
    2. Convertire 101,011₂ in decimale

## Analisi degli errori

* + 1. Calcolare la rappresentazione binaria di 0,1₁₀ e confrontarla con la sua approssimazione in IEEE 754
    2. Dimostrare perché 0,1 + 0,2 ≠ 0,3 esattamente nei computer

## Rappresentazioni speciali

* + 1. Rappresentare +∞ in formato IEEE 754 single precision
    2. Rappresentare NaN in formato IEEE 754 single precision

# Conclusioni

 La rappresentazione binaria dei numeri decimali è fondamentale nell'informatica  Lo standard IEEE 754 risolve molti problemi pratici, ma ha limitazioni intrinseche

 La comprensione degli errori di arrotondamento è cruciale per lo sviluppo di software affidabile

 Per calcoli che richiedono precisione assoluta, esistono alternative come aritmetica a precisione arbitraria o tipi di dati specializzati

**Esercizi sulla Rappresentazione Binaria dei Numeri Decimali e in Virgola Mobile**

**Sezione 1: Conversioni Base (Livello Base)**

**Esercizio 1.1**

Converti i seguenti numeri decimali interi in binario:

1. 27
2. 45
3. 64
4. 102
5. 255

**Esercizio 1.2**

Converti i seguenti numeri binari in decimale:

1. 10101
2. 11100
3. 1000000
4. 10001001
5. 11111111

**Esercizio 1.3**

Converti le seguenti frazioni decimali in binario (fino a 8 bit dopo la virgola):

1. 0,5
2. 0,25
3. 0,75
4. 0,375
5. 0,4

**Esercizio 1.4**

Converti i seguenti numeri binari frazionari in decimale:

1. 0,1
2. 0,01
3. 0,101
4. 0,0011
5. 0,10101

**Sezione 2: Numeri Misti (Livello Intermedio)**

**Esercizio 2.1**

Converti i seguenti numeri decimali in binario:

1. 13,625
2. 5,75
3. 27,125
4. 42,0625
5. 9,9

**Esercizio 2.2**

Converti i seguenti numeri binari in decimale:

1. 101,11
2. 1010,101
3. 110,0101
4. 11,111
5. 1000,001

**Esercizio 2.3**

Determina quali dei seguenti numeri decimali hanno una rappresentazione finita in binario:

1. 0,5
2. 0,1
3. 0,75
4. 0,2
5. 0,625

**Sezione 3: Rappresentazione in Virgola Mobile (Livello Avanzato)**

**Esercizio 3.1**

Rappresenta i seguenti numeri in formato IEEE 754 a singola precisione (32 bit). Mostra i bit del segno, esponente e mantissa:

1. 10,5
2. -27,25
3. 0,125
4. 64
5. -0,1

**Esercizio 3.2**

Converti le seguenti rappresentazioni IEEE 754 a singola precisione in numeri decimali:

1. 0 10000010 01000000000000000000000
2. 1 10000001 10010000000000000000000
3. 0 01111110 00000000000000000000000
4. 1 00000000 00000000000000000000000
5. 0 11111111 00000000000000000000000

**Esercizio 3.3**

Calcola manualmente il risultato delle seguenti operazioni in binario:

1. 101,01 + 11,1
2. 1010,1 - 101,11
3. 11,01 × 10,1
4. 110,01 ÷ 10,1

**Sezione 4: Errori di Arrotondamento (Livello Avanzato)**

**Esercizio 4.1**

Calcola e confronta:

1. Rappresenta 0,1 in binario (fino a 10 cifre dopo la virgola)
2. Rappresenta 0,2 in binario (fino a 10 cifre dopo la virgola)
3. Somma le rappresentazioni binarie di 0,1 e 0,2
4. Confronta il risultato con la rappresentazione binaria di 0,3
5. Spiega la discrepanza

**Esercizio 4.2**

Per ciascuno dei seguenti numeri, indica se può essere rappresentato esattamente in IEEE 754 e spiega perché:

1. 0,5
2. 0,1
3. 0,333...
4. 2^(-10)
5. 10^(-5)

**Sezione 5: Casi Speciali IEEE 754 (Livello Avanzato)**

**Esercizio 5.1**

Rappresenta i seguenti valori speciali in IEEE 754 a singola precisione:

a) +0

b) -0

c) +∞

1. -∞
2. NaN

**Esercizio 5.2**

Determina quale valore speciale IEEE 754 risulta dalle seguenti operazioni:

1. 1.0 / 0.0
2. 0.0 / 0.0
3. √(-1)
4. log(0)
5. ∞ - ∞

**Sezione 6: Problemi Applicativi (Livello Avanzato)**

**Esercizio 6.1**

Analisi di precisione:

1. Calcola il più piccolo numero positivo rappresentabile in IEEE 754 singola precisione
2. Calcola il più grande numero finito rappresentabile in IEEE 754 singola precisione
3. Qual è la precisione (in cifre decimali) della singola precisione IEEE 754?
4. Qual è la precisione (in cifre decimali) della doppia precisione IEEE 754?
5. Perché i computer usano la base 2 invece della base 10 per rappresentare i numeri?

**Esercizio 6.2**

Algoritmi:

1. Implementa un algoritmo per convertire un numero decimale in binario (pseudocodice)
2. Implementa un algoritmo per convertire un numero binario in decimale (pseudocodice)
3. Descrivi un algoritmo per sommare due numeri in virgola mobile in IEEE 754
4. Come gestiresti i casi di overflow e underflow in un'implementazione software?
5. Proponi una soluzione per eseguire calcoli decimali esatti senza errori di arrotondamento

**Esercizio 6.3**

Applicazioni pratiche:

1. Un sistema bancario necessita di gestire importi monetari con precisione al centesimo. Quale rappresentazione numerica sarebbe più appropriata e perché?
2. In un sistema di navigazione GPS, quali problemi potrebbero verificarsi a causa degli errori di arrotondamento?
3. Un software scientifico deve calcolare il valore di π con alta precisione. Quale approccio utilizzeresti?
4. Spiega perché l'espressione nei linguaggi di programmazione spesso restituisce

false

0.1 + 0.2 == 0.3

1. Progetta un sistema che minimizzi gli errori di arrotondamento per una serie di calcoli finanziari