Alexandria University

Faculty of Engineering

Communication and Electronics Department

DS Sheet 1

Primitive Data Types and Arrays

Name/ Mostafa Ahmed Mohamed Rashed
ID/ 19017528
Section/ 3

1. **(Floating Points)** Floating-point numbers in a certain hexadecimal computer are to be represented using 48-bits: One for the sign, seven for the characteristic part, and forty for the mantissa.

1. What is the precision in such representation?
2. What is the largest and least positive numbers that can be represented in such system?
3. What is the limit of the relative chopping error that may be introduced in representation of a real data item?
4. How is the zero represented?

Solution

1 Sign	7 characteristic	40 Mantissa

$\because$

Hexadecimal computer => R = 16

$\because$ Mantissa is 40-bit long => N = 10 hex numbers

$\because$ characteristic is 7-bit long

$\therefore 0 \leq$ biased exp. $\leq 2^7$

$\therefore -64 \leq$ true exp. ≤ 63 => $E_{min} = -64$, $E_{max} = 63$

1. Precision = 40/4 = 10
2. $\therefore$ Largest positive number = $(1-R^{-n})*R^{Emax}$
   $= (1-16^{-10})*16^{63} = 7.23700558\times10^{75}$
   $\therefore$ Least positive number = R^{Emin-1}
   $= 16^{-64-1} = 16^{-65} = 5.39760535\times10^{-79}$
3. Limit of relative chopping error = $R^{-(n-1)}$
   $= 16^{-(10-1)} = 16^{-9} = 1.45519152\times10^{-11}$
4. Zero is represented by 12 zeroes and in the de-normalized form with characteristic and Mantissa equal zero the sign bit could be 0 indicating +0 or 1 indicating -0 they are both distinct values but compare as equal
   ⇨ $0.00\text{x}16^{-64}$

2. **(Relative Error)** Which of the following floating-point systems has the least bound of relative error in the representation of real values :

1. A system with 6 Hexadecimal digits $= 16^{-5}$
2. A system with 7 Decimal digits $= 10^{-6}$
3. A system with 8 Octal digits $= 8^{-7}$

$\because \frac{16^{-5}}{8^{-7}}$ & $\frac{10^{-6}}{8^{-7}} > 1$

$\because$ Bound relative error = $R^{-(n-1)}$

$\therefore$ System with 8 octal digits has the least bound of relative error

3. **(Array Mapping)** Given the array `X[L1..U1, L2..U2, L3..U3, L4..U4, L5..U5];` each element in the array occupies 2 memory cells. Derive the appropriate addressing equation for an element that has the indexes `s1,s2,s3,s4,s5` and can be accessed as `X[s1][s2][s3][s4][s5],` if X is stored:

1. in a row-major order
2. in a column-major order

$\because$ C = 2

1. Loc(X[s1][s2][s3][s4][s5]) = Loc(X[L1][L2][L3][L4][L5]) + 2 * {
   (U5-L5+1) (U4-L4+1) (U3-L3+1) (U2-L2+1) (s1-L1)
   + (U5-L5+1) (U4-L4+1) (U3-L3+1) (s2-L2)
   + (U5-L5+1) (U4-L4+1) (s3-L3)
   + (U5-L5+1) (s4-L4)
   + (s5-L5)}

2. Loc(X[s1][s2][s3][s4][s5]) = Loc(X[L1][L2][L3][L4][L5]) + 2 * {
   (U1-L1+1) (U2-L2+1) (U3-L3+1) (U4-L4+1) (s5-L5)
   + (U1-L1+1) (U2-L2+1) (U3-L3+1) (s4-L4)
   + (U1-L1+1) (U2-L2+1) (s3-L3)
   + (U1-L1+1) (s2-L2)
   + (s1-L1)}

4. **(Sparse Matrices)** If "S" is a pxq matrix with "k" nonzero elements, for what values of "k" does the coordinates-method use less storage space than "S"? (Assume that each of the coordinates occupy the same amount of space as an element of "S").

let S in 2-D representation stores => p*q*c , where c is the amount of space

co-ordinate matrix occupies => 3*k*c

Assume: 3*k*c < p*q*c, $\therefore k < \frac{p*q}{3}$

5. **(Sparse Matrices)** Assume a sparse matrix X of size m x n is to be saved. The array X is estimated to have a maximum of p% nonzero elements. Each array element takes c memory cells. The number of bits required per cell is b.

1. Find the ratio between the memory spaces required to save X as a 2D-array and using the Coordinate method.
2. For what values of p does the coordinate method use less storage than the 2D-array representation?

2-D array =>m*n*c

Co-ordinate method bitmap requires row, column, V each of length equal V => 3*{(m*n)*(p/100)}*c

1. m*n*c : {3*{(m*n)*(p/100)}*c} ----------------→ ÷ m*n*c

   $1 : \frac{3*p}{100}$

2. Assume : 3*{(m*n)*(p/100)}*c < m*n*c => $\frac{3*p}{100} < 1$, $3*p < 100$,

   $p < \frac{100}{3}$

6. **(Triangular Matrices)** Derive the mapping function required to map between the indexes I and j of a lower triangular matrix (represented as 2D array) and the index k of the more efficient linear row-major representation of this matrix. State the range of k. Repeat for the column-major representation for symmetric matrices.

Assuming first element in array is of index 1

- 2-D mapping to row major of lower triangular matrix:
  [1][2,3][4,5,6]
  k = 1+ 2 + 3 + … + (i - 1) + j
  k = (i*(i-1) / 2) + j
  min k is when i=j=0 => k=0
  max k is when i=j=n => k= (n*(n-1) / 2) + n = (n*(n+1)) /2

1		
2	3	
4	5	6

- 2-D mapping to column major of symmetric matrix:
  - Assume <u>lower triangular</u> and use column major
    [1,2,3][4,5][6]
    Let matrix is of size n
    k = n + (n-1) + (n-2) + … + (n-(j-2)) + (i-j+1)
    k = {n*(j-1) - (1+2+3+…+(j-2))} + (i-j+1)
    k = (n*(j-1) - $\frac{(j-2)(j-1)}{2}$) + (i-j+1)
  - Assume <u>upper triangular</u> and use column major
    [1,2,3][4,5][6]
    Same as row major of lower triangular matrix
    k = 1+ 2 + 3 + … + (j - 1) + i
    k = (j*(j-1) / 2) + i

1	2	3
2	4	5
3	5	6

7. **(Triangular Matrices)** Write an algorithm to calculate the sum of two triangular matrices A and B.

SOL1: Loop through indices of both matrices and add each element

```
 for (i = 0; i < n_rows; i++){
   for (j = 0; j < n_columns; j++) {
     sum[i][j] = a[i][j] + b[i][j];}}
```

SOL2: Map the vectors using column major then add each element

```
for (k = 0; k < (n*(n-1) / 2) ; k++){
sum[k] = a[k] + b[k];}
```