Ch. 2 Array 參考答案

Question 1

A hacker is transmitting a virus through the internet. The virus has a special pattern denoted as P, and the value is "11101101". Now we want to detect whether a downloaded file includes this kind of virus.

(a) We are going to use the KMP algorithm to detect, and need to build a failure function. Please fill out the blank below.

P[i]	1	1	1	0	1	1	0	1
F(i)	-1	0						

(b) Please try to write a program to calculate the failure function.

(c) Please evaluate the big-O complexity of this program.

(Question 1) Ans:

a.

P[i]	1	1	1	0	1	1	0	1
F(i)	-1	0	1	-1	0	1	-1	0

b. (Please refer to the textbook)

c. O(n)

Question 2

In the class, we introduced two kinds of polynomial representation (sparse and dense). What are the advantages and disadvantages of these two methods from the aspect of memory usage and runtime.

(Question 2) Ans:

- a, b: two polynomials with max degree=D
- a.terms=n, b.terms=m
- When a,b are sparse, n,m << D; otherwise, n,m = D.

	Dense	Sparse
Time	D	n+m
Memory	2D	2(n+m)
Implementation	Easier	Harder

Question 3

Compare the following two versions of Matrix Multiplication.

a) Compare the time complexity of the two Methods.

b) Which version should run faster when the size of the matrix is large?

Method 1 :

Method explained in the video: $c[i][j] = \sum_{k=0}^{n-1} a[i][k] \times b[k][j]$

```
Matrix Matrix::Multiply(Matrix b){
    Matrix c(rows, b.cols);
    for (i=0; i<rows; i++) {        // O(rows)
        for (j=0; j<b.cols; j++) { // O(b.cols)
            sum=0;
            for (k=0; k<cols; k++)  // O(cols)
                sum += a[i][k]*b[k][j];
            c[i][j]=sum;
        }
    }
    return c;
}
```

Method 2:

Let Matrix t = transpose of Matrix b, then calculate $c[i][j] = \sum_{k=0}^{n-1} a[i][k] \times t[j][k]$

```
Matrix Matrix::Multiply(Matrix b){
    Matrix c(rows, b.cols);
    Matrix t = b.transpose();
    for (i=0; i<rows; i++) {        // O(rows)
        for (j=0; j<b.cols; j++) { // O(b.cols)
            sum=0;
            for (k=0; k<cols; k++)  // O(cols)
                sum += a[i][k]*t[j][k];
            c[i][j]=sum;
        }
    }
    return c;
}
```

(Question 3) Ans:

a) both $O(n^3)$

b) When n is large, method 2 should run faster. (Due to the fact that arrays in c++ are stored in row-major format in memory. https://www.appentra.com/knowledge/checks/pwr010/)

Question 4

Consider the sparse matrix A as shown below. Suppose that each non-zero element in the matrix is stored by a triple <row, column, value>. And all the triples are stored in an array in the row-major order.

$$A = \begin{bmatrix} 0 & 4 & 0 \\ 0 & 5 & 0 \\ 6 & 0 & 0 \end{bmatrix}$$

a) Draw the array that represents matrix A.

Consider the fast transpose algorithm. The goal is to obtain the transpose of sparse matrix A step by step.

b) Write down the array that stores <rowSize, rowStart> pairs for all rows in A^T.

c) Write down the array that represents A^T (in which non-zero elements of A^T are represented by <row, column, value> triples).

(Question 4) Ans:

a) [(0,1,4), (1,1,5), (2,0,6)]

b)

column	rowsize	rowstart
0	1	0
1	2	1
2	0	3

c)

	row	column	value
[0]	0	2	6
[1]	1	0	4
[2]	1	1	5

Question 5

a) Explain what a sparse matrix is?

b) Please calculate the approximate memory requirements of the matrices in the following for the two given methods and tell which one is more efficient.

	2D Array	Class SparseMatrix (講義 P26)	Which is more efficient?
(a) $\begin{bmatrix} 2 & 0 & 4 \\ 6 & 7 & 33 \\ 5 & 0 & -2 \end{bmatrix}$			
(b) $\begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \end{bmatrix}$ (a~t are all non-zero elements)			
(c) 2000 x 2000 matrix with 200 nonzero elements			

(Question 5) Ans:

	2D-Array	SparseMatrix	Better?
(a)	9 integers	7*3 integers	2D-Array
(b)	20 integers	20*3 integers	2D-Array
(c)	2000*2000 integers	200*3 integers	SparseMatrix

Question 6

a) When do we use Sparse Matrix to store a matrix ?

b) If we want to store the matrix below, what shall smArray contain?

1	0	3	0	1
0	0	0	1	0
0	0	1	0	0
0	0	0	1	1

```
class MatrixTerm{
    friend class SparseMatrix;
     private:
         int row, col, value;
};

class SparseMatrix{
    private:
        int rows, cols, terms, capacity;
        MatrixTerm *smArray;
}
```

(Question 6) Ans:

a) When there are a lot of zero terms.

b) [(0,0,1), (0,2,3), (0,4,1) , (1,3,1), (2,2,1) , (3,3,1), (3,4,1)]

Question 7

The time complexity of a 2-dimensional matrix multiplication is O(n^3). The computation cost is high for multiplication of two really large matrices. Please discuss whether the order of the matrix multiplication influence the computing time. For example, you may try to change the order of $M_1 * M_2$ to $M_2 * M_1$ and compute the result.

(Question 7) Ans:

Say $M_1: 3 * 7$, $M_2: 7 * 3$

$M_1 * M_2$: 7 (multiplications + addition) * 3^2 output elements

$M_2 * M_1$: 3 (multiplications + addition) * 7^2 output elements

So the order of matrices does matter.

Question 8

Consider the following declaration of a two-dimensional array in C++: Assuming that the main memory is byte-addressable and that an integer (4-byte) array for A[n][m] is stored starting from memory address 0, what is the address of element A[40][50](n>41, m>51) ? There are two possible answers based on different storage method. (Row-major and column-major)

(Question 8) Ans:

Row-major: 40m+50

Column-major: 50n+40

Question 9

We need to build a multiplier to compute the power of 11, but the output of the multiplier, for example 11^{100}, often exceeds the valid integer range (-2,147,483,648 to 2,147,483,647). Please try using the array representation to deal with this situation.

- **Large Number manipulation:** https://www.geeksforgeeks.org/multiply-large-numbers-represented-as-strings/
- **Note:**
  1. **Store the number as a string array.**
  2. **Convert parts of the string into integer.**
  3. **Do calculation on the converted part, and store the outcome as a partial result.**
  4. **Repeat 2 and 3 until all parts are done, then combine all of the partial results into a final result.**