Two-Dimensional Arrays

Jyh-Shing Roger Jang (張智星)
CSIE Dept, National Taiwan University

Dynamically Allocated Matrices

- How to create and delete a dynamically allocated matrix of n by m:
  - Create

```
int** M = new int*[n];              // allocate an array of row pointers
for (int i = 0; i < n; i++)
  M[i] = new int[m];                // allocate the i-th row
```

A 2D matrix of 3x5

  - Delete

```
for (int i = 0; i < n; i++)
  delete[] M[i];                    // delete the i-th row
delete[] M;                         // delete the array of row pointers
```

- Using STL vectors instead
  - Declare

```
vector< vector<int> > M(n, vector<int>(m));
```

  - Delete: do nothing

Two Ways to Compute Sum of Elements in a 2D Matrix

Quiz: Which is faster? Why?

Link to code

Add compiler optimization option: -O0, -O2, -O3, -Ofast

Why do we need to specify the column size?

- Row sum first

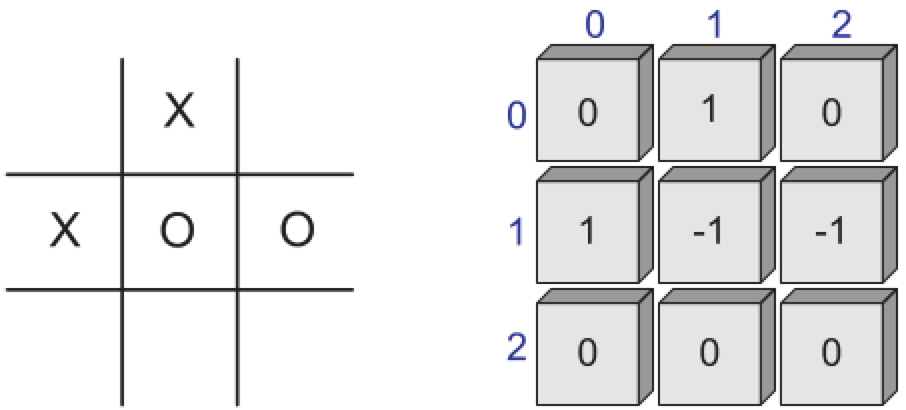
```c
// Calculate row sum first
int rowSum(int array[][COL]){
  int sum=0;
  for(int i=0; i<ROW; i++)
    for(int j=0;j<COL;j++)  // row sum
      sum+=array[i][j];
  return sum;
}
```

- Column sum first

```c
// Calculate column sum first
int colSum(int array[][COL]){
  int sum=0;
  for(int j=0; j<COL; j++)
    for(int i=0;i<ROW;i++)  // col sum
      sum+=array[i][j];
  return sum;
}
```

Example: Game of Tic-Tac-Toe

- Examples
  - ticTacToe00.cpp



playing board board array

- Observation
  - It quite easy to implement it as an interactive game.

FAQ

- What is stack and heap?
- Segmentation fault on large array sizes