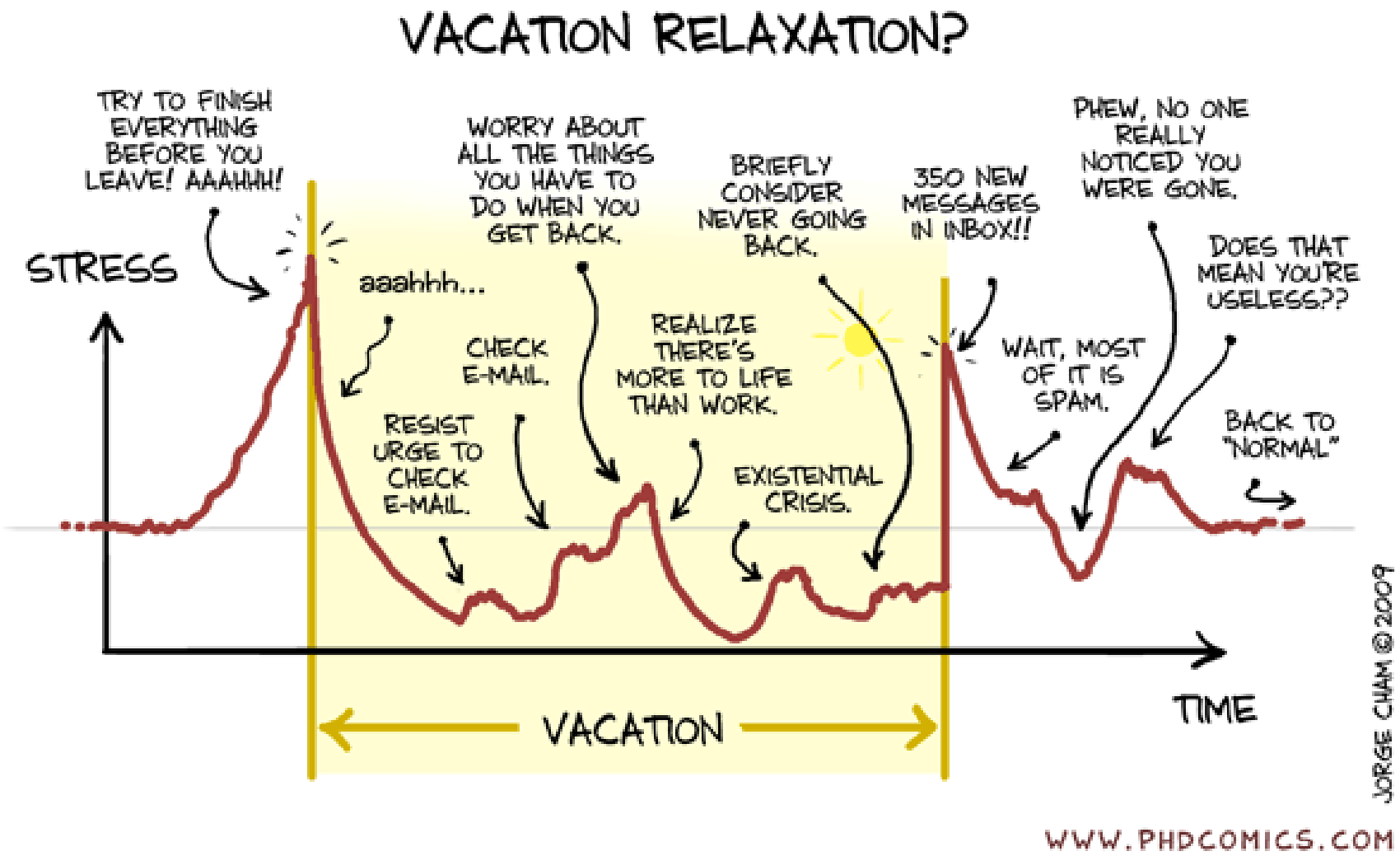
CS 161, Lecture 19: Multidimensional Arrays – 26 February 2018



Minor Correction

- Char arrays which are null terminated are C-style strings
  - That means you can use functions from <cstring> (string.h) on them
- Char arrays do not need to be null terminated
  - Just means you can't use <cstring> (string.h) on them
- We mostly default to C-style strings and expect null terminator on character arrays because most problems we solve involving characters are strings
- Two types of strings: counted (C++) and null terminated (C)

Multidimensional Arrays

- data_type array_name[rows][cols];
  - int mult_table[5][5];
  - char cross_word[15][15]
  - float grades[num_students][num_grades]
- Example of multidimensional problem:
  - 2D
  - 3D
  - 4D

Initializing 2D Arrays

- Declaration: int arr[2][3] = {{0,0,0},{0,0,0}}
- Individual Elements:
  - arr[0][0] = 0;
  - arr[0][1] = 0;
  - arr[0][2] = 0;
  - arr[1][0] = 0;
  - arr[1][1] = 0;
  - arr[1][2] = 0;
- Loop

```
for(int i=0; i<2; i++) {
        for(int j=0; j<3; j++) {
                arr[i][j] = 0;
        }
}
```

Reading and Printing 2D Arrays

- Reading

```
for(int i=0; i<2; i++) {
        for(int j=0; j<3; j++) {
                cout << “Enter a value: “;
                cin >> arr[i][j] ;
        }
}
```

Printing:

```
for(int i=0; i<2; i++) {
        for(int j=0; j<3; j++) {
                cout << arr[i][j] << “ “;
        }
        cout << endl;
}
```

Dynamic 2D Arrays

- 1D dynamic array

- 2D dynamic array

Declaration, Initialization and Deletion

```
int ** ar;
ar = new int*[rows];
for(int i=0; i<rows; i++) {
        ar[i] = new int[cols];
}

for(int i=0; i<rows; i++) {
        delete [] ar[i];
}
delete [] ar;
```

Demo