

Welcome
to CS 314!





Your TA (me!)

- Skyler Vestal
- CS + Math Junior ('23)
- Completed a lot of stuff. Ask if you have questions about other CS/Math topics!
- Interning at Zillow this summer (Seattle)!
- skylervestal@utexas.edu

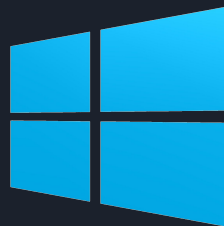


Also me!

- #1 Brewers Fan!
- Big Packers Fan
- Rollerblading
- Learning Japanese (N3-ish)
- Windows & Linux
- VS Code Simp



日本語



Dinoswar

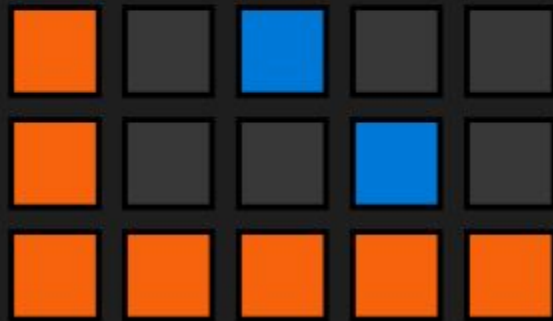
55.5K subscri

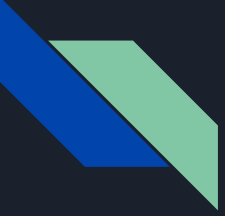
This channel
analyzing, ...





Wordle 226 3/6





Some links!

- [Style Guide](#)
- [My Website](#)

Discussion Section Name

MUTE YOUR
MOKE





Algorithm Analysis!


```
public void removeRange(int start, int stop) {  
    if (stop < start || start < 0 || stop > size) {  
        throw new IllegalArgumentException("incorrect range");  
    }  
    if (stop > start) {  
        int numRemoved = stop - start;  
        for (int i = stop; i < size; i++) {  
            con[i - numRemoved] = con[i];  
        }  
        size -= numRemoved;  
    }  
}
```

1. A method is $O(2^N)$. It takes 1 second for the method to complete when $N = 50$.

What is the expected time in seconds for the method to complete when $N = 54$?

$$\frac{2^{54}}{2^{50}} = \frac{x}{1}$$

LATEX

2. A method is $O(N^2 \log_2 N)$. It takes 20 seconds for the method to complete when $N = 1,000,000$.

What is the expected time in seconds for the method to complete when $N = 2,000,000$?

$$\frac{2 \log_2(2 \times 10^6)}{\log_2(1 \times 10^6)} = \frac{x}{20}$$

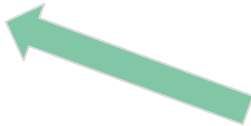
3. What is the $T(N)$ for method a? Recall, $T(N)$ is the function that represents the *actual* number of executable statements for a function or algorithm. $N = \text{listA.length} = \text{listB.length}$

```
// pre: listA.length == listB.length
public int a(int[] listA, int[] listB) {
    int total = 0;
    for (int i = 0; i < listA.length; i++)
        for (int j = 0; j < listB.length; j++) {
            total += listA[i] * listB[j];
            total += listA[i] / 10;
            total += listB[j] / 100;
        }
    return total;
}
```

$$1 + 1 + N + 1 + N + N(1 + N + 1 + N + N(1 + 1 + 1)) + 1$$

4. What is the worst case order (Big O) of method `d`? Assume `Arrays.fill` is $O(N)$ and method `process` is $O(N)$. $N = \text{data.length}$

```
public int[] d(int[] data, int key) {  
    int[] result = new int[0];  
    for (int i = 0; i < data.length; i++) {  
        if (data[i] == key) {  
            result = new int[data.length];  
            Arrays.fill(data, i);  
            process(data, i, key);  
        }  
    }  
    return result;  
}
```



5. What is the best cases order of the following method? Assume method `numRows` is $O(1)$ and that method `process` is $O(1)$. Assume method `numRows` returns the number of rows in the 2d array sent as a parameter.

// mat is a square matrix. All rows have `mat.length` columns.

```
public static int num6(int[][] mat) {  
  
    int result = 0;  
    for (int r = 0; r < numRows(mat); r++) {  
        int[] row = mat[r];  
        for (int c = 0; c < mat[0].length; c++) {  
            int val = mat[r][c];  
            result += process(val, mat, row);  
        }  
    }  
    return result;  
}
```

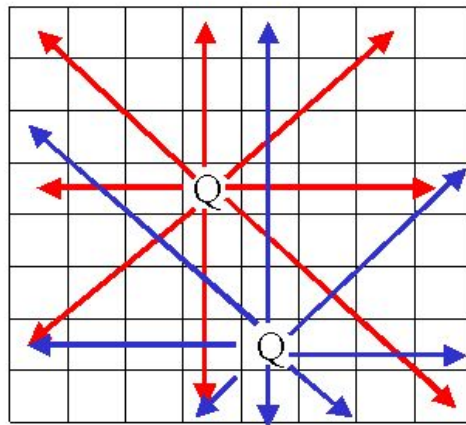
9. What is output by the following code? (it is very gacky)

```
int n = 64;  
int r = 0;  
for (int i = 1; i <= n; i *= 2) {  
    for (int j = 1; j <= i; j++) {  
        r++;  
    }  
}  
System.out.print(r);
```

$$1 + 2 + 4 + \dots + N/2 + N = 2N - 1$$

QueensAreSafe

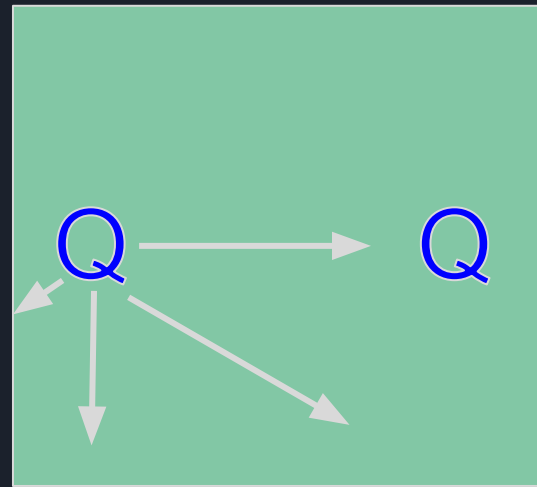
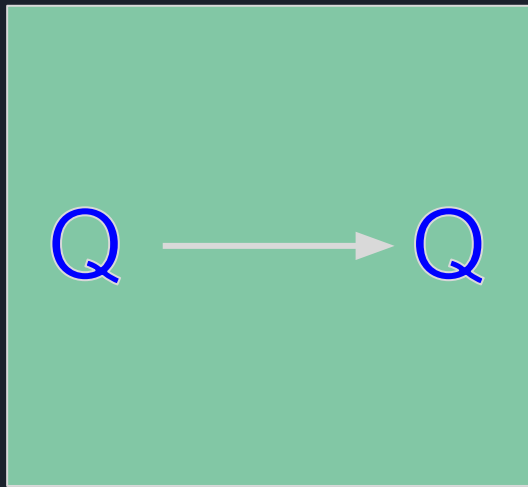
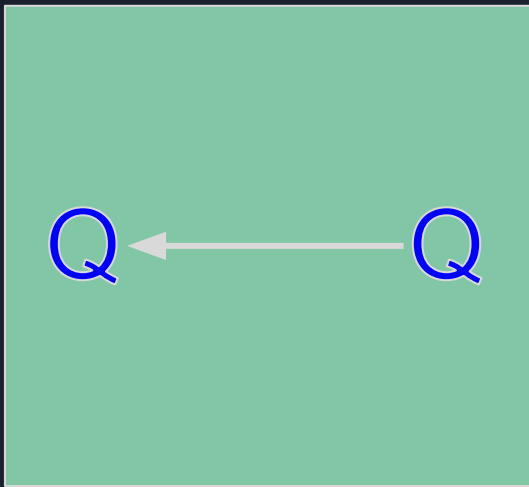
- Relevant Directions
- Parameterized Row/Col Solutions
- Slope Method





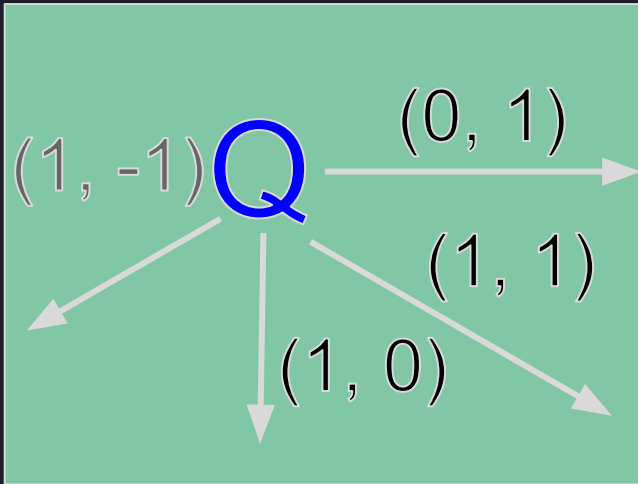
Directions

- Only need to check 4 directions



Directions

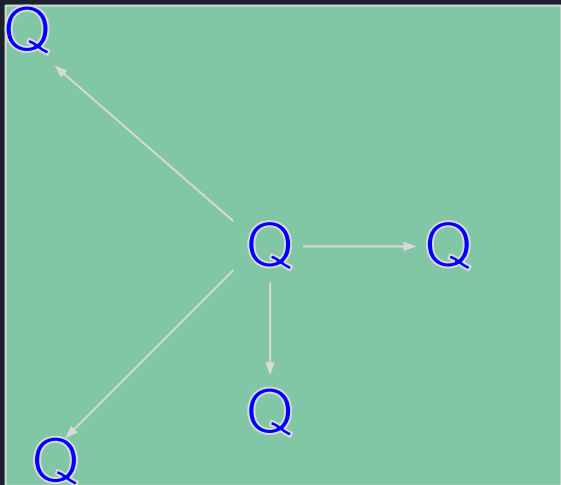
- You can use an array to store the different changes in rows and columns



rows = {0, 1, 1, 1}
cols = {1, 1, 0, -1}

Directions

- You can also use the slopes between queens to determine if they're in a line (if the slope is 0 or 1)

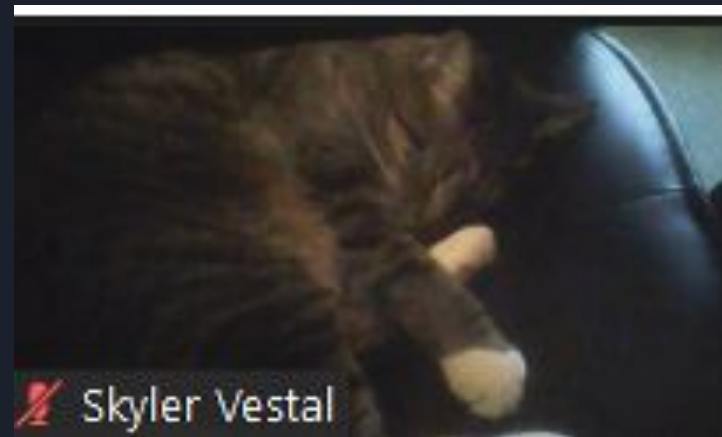


$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\textit{rise}}{\textit{run}}$$



Grading Schedule

- Grading on Saturday
- May spill to Sunday
- Some exceptions ...





Assignment Grading

- Don't email me for:
 - Disagreeing on taking of a point for something you did
- *Please* email me for:
 - Mistake with your correctness
 - Mistake with adding up grade
 - Inconsistent deduction w/ past assignment
 - Inconsistency w/ assignment page
 - I took off for something you didn't do



Assignment Grading

- I can only regrade for the five days after I release grades
- Don't get stressed about small **style** deductions:
 - A single exam coding Q is a little less than an entire assignment
 - Assignments only make up 22% of your grade
 - Y'all get 40 slack points + 10 for extra credit
 - I lost 10 points on my first 3 assignments and had slack points to spare (we had ~25)



Common Style Issues

- Spacing on operators (**AUTO FORMATTER!!!!!!!!**):
 - `3+3` -> `3 + 3`
 - `if(...)` -> `if (...)`
 - `public int method ()` -> `method() {`
 - `//test` -> `// test`
- Lines should be 100 long (set a vertical line)
- Private instance variables
- Checking preconditions



Common Style Issues

- ***USE AN AUTO FORMATTER!!!!!!!***
- ***USE AN AUTO FORMATTER!!!!!!!***
- ***USE AN AUTO FORMATTER!!!!!!!***
- ***USE AN AUTO FORMATTER!!!!!!!***
- ***USE AN AUTO FORMATTER!!!!!!!***



Common Style Issues

- Magic numbers:
 - BAD:
 - `if (year < 10)`
 - GOOD:
 - `final int PERIOD_LENGTH = 10;`
 - `if (year < PERIOD_LENGTH)`
 - If you're using a magic number in multiple methods, declare it at the top of your class.



Common Style Issues

- Returning early:

// BAD:

```
int sum = 0;
for (int i = 0; i < a.length; i++) {
    if (a[i] == 0) {
        sum += 1;
    }
}
return sum == 0;
```

// GOOD:

```
for (int i = 0; i < a.length; i++) {
    if (a[i] == 0) {
        return false;
    }
}
return true;
```



Common Style Issues

- Boolean zen (part 1):

```
// BAD:  
if (a == 0) {  
    return true;  
} else {  
    return false;  
}
```

```
// GOOD:  
return a == 0;
```



Common Style Issues

- Boolean zen (part 2):

```
// BAD:  
if (val == true) {  
    ...  
}
```

```
// GOOD:  
if (val) {  
    ...  
}
```



Common Style Issues

- Preferred method header comments:

```
// Calculates the amount of birds in my yard at a given time  
// pre: bar != null, t >= 0  
// post: returns birds at time t  
public int foo(int[] bar, int t) {
```

```
// Prints the amount of snails on my desk  
// pre: none (For this example, bar handles null vals)  
// post: none  
public void bar(String desk) {
```



Style Preferences

- I can't take off for this, but I'd prefer:

```
public void foo(String desk) {  
    if (a) {  
  
    }  
}
```

```
// rather than  
public void foo(String desk)  
{  
    if (a)  
    {  
  
    }  
}
```



Common Matrix Issues

- Style:
 - Write the preconditions that Mike specifies in header comments!
 - Return early in equals if possible
- Experiments
 - 1 int = 4 bytes
 - The Big O you report should be based off your algorithm analysis
 - We want to see if your timing supports this! Be honest!!! It doesn't matter some are fuzzy/seem off.