

Required: Student Setup

As the first part of your homework, follow the instructions below to get your environment properly set up for this class. This section is not graded but you will need to do it in order to submit your programming assignments.

- Email fsareshwala@berkeley.edu with a request to be invited to the Campuswire class if you don't already have access
- Complete the instructions on <https://github.com/fsareshwala/cs404.1/blob/master/student-setup.md> to set up your programming and development environment

Optional: Student Survey

Once you have set up your environment, you will find a file called `survey.md` in the top-level root of the repository. This file contains a survey that asks various questions about your experience with computer science, software engineering, and data structures and algorithms. Answering these questions is completely optional and doesn't affect your grade on this or any future assignment. However, your responses do help me understand your current experience and better tailor the course to your individual needs.

If you are comfortable with giving me a bit more information, please complete the survey within the `survey.md` file in the top-level root of your repository. Your responses will be kept confidential and will never be shared. Once you are done, commit your responses and push them to your upstream repository.

Version Control Systems: `git`

1. What are some of the reasons to use version control systems?

.....

.....

.....

.....

2. What advantages does `git` have over previous generations of version control systems?

.....

.....

.....

.....

3. You get the following error when working with `git`:

```
Pushing to https://github.com/fsareshwala/dotfiles
To https://github.com/fsareshwala/dotfiles
! [rejected]        master -> master (fetch first)
```

What went wrong?

.....

.....

4. Files in the directory `program1` are currently untracked. How can you get `git` to track them? You may write either the conceptual ideas or the commands necessary that will have `git` track them.

.....

Complexity

5. Suppose that $T_1(n) = O(f(n))$ and $T_2(n) = O(f(n))$. Which of the following are true:

A. $T_1(n) + T_2(n) = O(f(n))$

B. $T_1(n) - T_2(n) = O(f(n))$

C. $\frac{T_1(n)}{T_2(n)} = O(1)$

D. $T_1(n) = O(T_2(n))$

6. Group the following into equivalent Big-O functions: n^2 , n , $n^2 + n$, $n^2 - n$, and $\frac{n^3}{n-1}$.

.....

7. Programs A and B are analyzed and are found to have worst case running times no greater than $150n \log n$ and n^2 , respectively. Answer the following questions.

(a) Which program has the better guarantee on the running time for large values of n ($N > 10,000$)?

.....

(b) Which program has the better guarantee on the running time for small values of N ($N < 100$)?

.....

(c) Which program will run faster on average for $N = 1,000$?

.....

(d) Can program B run faster than program A on all possible inputs?

.....

8. In terms of n , what is the running time of the following algorithm to compute x^n ?

```
double power(double x, int n) {
    double result = 1.0;
    for (int i = 0; i < n; i++) {
        result *= x;
    }
    return result;
}
```

.....

9. An algorithm takes 0.5 ms for an input size of 100 elements. How long will this algorithm take for an input size of 500 elements (assuming that lower order terms are negligible) if the running time is:

(a) Linear: $O(n)$

.....

(b) Linearithmic: $O(n \log n)$

.....

(c) Quadratic: $O(n^2)$

.....

(d) Cubic: $O(n^3)$

.....

10. For each of the following program fragments, analyze the running time and provide a time complexity in Big-O notation.

(a) `for (int i = 0; i < n; i++) {
 sum++;
 }`

.....

(b) `for (int i = 0; i < n; i += 2) {
 sum++;
 }`

.....

(c) `for (int i = 0; i < n; i++) {
 for (int j = 0; j < n; j++) {
 sum++;
 }
 }`

.....

(d) `for (int i = 0; i < n; i++) {
 sum++;
 }

 for (int j = 0; j < n; j++) {
 sum++;
 }`

.....

(e) `for (int i = 0; i < n; i++) {
 for (int j = 0; j < n * n; j++) {
 sum++;
 }
 }`

.....

(f) `for (int i = 0; i < n; i++) {
 for (int j = 0; j < i; j++) {
 sum++;
 }`

```
    }  
}
```

```
.....  
(g)  for( int i = 0; i < n; i++) {  
      for(int j = 0; j < n * n; j++) {  
        for(int k = 0; k < j; k++) {  
          sum++;  
        }  
      }  
    }  
}
```

```
.....
```