

Natalie Craun  
p1\_answers

In big  $\Theta$  notation analyze the following function. Describe it as a function of the input size ( $n$ ).

```
int n; // Set to some non-zero value initially
int* A; // Points to an array of size n
int x = n;

// Suppose x & n are only changed by this function
1.  int f1(int* A)
2.  {
3.      if( x == 0) {
4.          x = n;
5.          return 1;
6.      }
7.      else if( (x % (int)sqrt(n)) == 0){
8.          for(int i=1; i <= n; i++){
9.              for(int j = 0; j < i; j++){
10.                 // do something O(1) with A[]
11.                 // x and n are unaffected
12.             }
13.         }
14.     }
15.     else {
16.         // do something O(1)
17.         // x and n are unaffected
18.     }
19.     x--;
20.     return 0;
21. }
```

The for loops in lines 8 and 9 run in  $\Theta(n^2)$ , but will only execute  $\frac{\sqrt{n}}{n}$  amount of the time because of the *else if* checking if  $x$  can be evenly divided by  $\sqrt{n}$  and the concept of amortized runtime, which requires us to divide the amount of times the condition holds true by  $n$ .

Therefore, the overall run time will be:  $(\frac{\sqrt{n}}{n} * n^2) = \Theta(n^{\frac{3}{2}})$