CS 61B Fall 2020

More Asymptotics

Exam Prep Discussion 7: October 5, 2020

1 Asymptotics is fun!

(a) Using the function `g` defined below, what is the runtime of the following function calls? Write each answer in terms of `N`.

```
1  void g(int N, int x) {
2      if (N == 0) {
3          return;
4      }
5      for (int i = 1; i <= x; i++) {
6          g(N - 1, i);
7      }
8  }
```

`g(N, 1)`: $\Theta($ $)$
`g(N, 2)`: $\Theta($ $)$

Solution:
`g(N, 1)`: $\Theta(N)$
`g(N, 2)`: $\Theta(N^2)$

(b) Suppose we change line 6 to `g(N - 1, x)` and change the stopping condition in the for loop to `i <= f(x)` where `f(x)` returns a random number between 1 and `x`, inclusive. For the following function calls, find the tightest Ω and big O bounds.

```
1  void g(int N, int x) {
2      if (N == 0) {
3          return;
4      }
5      for (int i = 1; i <= f(x); i++) {
6          g(N - 1, x);
7      }
8  }
```

`g(N, 2)`: $\Omega($ $)$, $O($ $)$
`g(N, N)`: $\Omega($ $)$, $O($ $)$

Solution:
`g(N, 2)`: $\Omega(N)$, $O(2^N)$
`g(N, N)`: $\Omega(N)$, $O(N^N)$

2 Flip Flop

For each part, give the best and worst case runtime in $\Theta(.)$ notation as a function of `N`. Your answer should be simple with no unnecessary leading constants or summations.

```
1  public static void flip(int N) {
2      if (N <= 100) {
3          return;
4      }
5      for (int i = 1; i < N; i++) {
6          // Assume g(i, N) will be equal to i for at least one i
7          if (g(i, N) == i) {
8              flop(i, N);
9              return;
10         }
11     }
12 }
```

Given the method `flip` defined above, we will determine the best and worst case runtime when `flop` is defined as:

```
1  public static void flop(int a, int b) {
2      flip(b - a);
3  }
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$

Solution:
Best Case: $\Theta(N)$, Worst Case: $\Theta(N)$

```
1  public static void flop(int a, int b) {
2      int low = Math.min(a, b - a);
3      flip(low);
4      flip(low);
5  }
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$

Solution:
Best Case: $\Theta(1)$, Worst Case: $\Theta(Nlog(N))$

```
1  public static void flop(int a, int b) {
2      flip(a);
3      flip(b - a);
4  }
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$

Solution:
Best Case: $\Theta(N)$, Worst Case: $\Theta(N^2)$

3 Prime Factors

What is the best and worst case runtime of the function below?

```
1  int prime_factors(int N) {
2      int factor = 2;
3      int count = 0;
4      while (factor * factor <= N) {
5          while (N % factor == 0) {
6              System.out.println(factor);
7              count += 1;
8              N = N / factor;
9          }
10         factor += 1;
11     }
12     return count;
13 }
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$

Solution:
Best Case: $\Theta(log(N))$, Worst Case: $\Theta(\sqrt{N})$