Asymptotic Notation and Analysis

4/4 questions correct

Excellent!

Retake

Next (/learn/data-structures-optimizing-performance/lecture/YBL2r/core-introduction-to-benchmarking)

✔ 1.

Consider the function $f(n) = 3\log_2(n) + 4n\log_2(n) + n$. What is the tightest big O class for $f(n)$?

- ○ O(log (n))
- ○ O(n log(n))

Well done!
Since $\log_2(n)$ and n each grow more slowly that $n\log_2(n)$, the term $4n\log_2(n)$ dominates both the term $3\log_2(n)$ and the term n. In asymptotic notation, keep only the dominant term and then drop the constants. Dropping the constants from $4n\log_2(n)$ gives $(O(\log(n))$.

- ○ O(n)

✔ 2.

What is the big-O class of the number of operations executed when running the method below, where n is the size of arr? Remember, when we're talking about Big-O analysis, we only care what happens when n is very large, so don't worry that this method might have problems on small array sizes.

```
int sumTheMiddle(int[] arr) {
  int range = 100;
  int start = arr.length/2 - range/2;
  int sum = 0;
  for (int i=start; i< start+range; i++)
  {
    sum += arr[i];
  }
  return sum;
}
```

- ○ $O(2^n)$
- ○ $O(n^2)$
- ○ $O(n)$
- ○ $O(1)$

Well done!
Good! We are starting at a constant less than the middle of the array, and going to a constant more.

✔ 3.

What is the big-O class of the number of operations executed when running the method below, where n is the size of arr? Remember, when we're talking about Big-O analysis, we only care what happens when n is very large, so don't worry that this method might have problems on small array sizes. [[Notice that the only difference between this method and the one from the previous question is the initialization of the variable "range".]]

```
int sumTheMiddle(int[] arr) {
  int range = arr.length/100;
  int start = arr.length/2 - range/2;
  int sum = 0;
  for (int i=start; i< start+range; i++)
  {
    sum += arr[i];
  }
  return sum;
}
```

- ○ $O(2^n)$

- O(n^2)
- O(n)

Well done!
Good! Now range is no longer a constant, so the size of the interval traversed by the for loop now depends on the length of the array.

- O(1)

✔ 4.

If a piece of code runs in linear time, O(n), what impact will doubling the input size have on the number of operations required to run?

- The number of operations will be roughly the same.
- The number of operations will roughly double.

Well done!

- The number of operation will grow, but you cannot say anything about by how much.
