Homework 2

January 25, 2012

1 Question 1

Solving a problem requires running an $O(N)$ algorithm, and then performing N binary searchers on an N-element array, and then running another $O(N)$ algorithm. What is the total cost of solving the problem?

2 Question 2

Order the following functions by growth rate: N, $\sqrt{N}$, $N^{1.5}$, N^2, $N \log N$, $N \log \log N$, $N \log^2 N$, $N \log(N^2)$, $2/N$, 2^N, $2^{N/2}$, 37, N^3, and $N^2 \log N$. Indicate which functions grow at the same rate.

3 Question 3

For each of the following program fragments, do the following:

1. Give a Big-O analyss of the running time.
2. Implement the code and run for several values of N.
3. Compare your analysis with the actual running times.

```
// Fragment 1
for (int i=0; i<n; i++)
  sum++;

// Fragment 2
for (int i=0; i<n; i+=2)
  sum++;

// Fragment 3
for (int i=0; i<n; i++)
  for (int j=0; j<n; j++)
    sum++;

// Fragment 4
for (int i=0; i<n; i++)
  sum++;
for (int j=0; j<n; j++)
  sum++;
```

```
// Fragment 5
for (int i=0; i<n; i++)
  for (int j=0; j<n*n; j++)
    sum++;

// Fragment 6
for (int i=0; i<n i++)
  for (int j=0; j<i; j++)
    sum++;

// Fragment 7
for (int i=0; i<n; i++)
  for (int j=0; j<n*n; j++)
    for (int k=0; k<j; k++)
      sum++;

// Fragment 8
for (int i=1; i<n; i=i*2)
  sum++;
```

4 Question 4

Occasionally, multiplying the sizes of nested loops can give an over-estimate for the Big-O running time. This result happens when an innermost loop is infrequently executed.

For the following program fragment, do the following:

1. Give a Big-O analyss of the running time.
2. Implement the code and run for several values of N.
3. Compare your analysis with the actual running times.

```
for (int i=1; i<=n; i++)
  for (int j=1; j<=i*i; j++)
    if (j%i == 0)
      for (int k=0; k<j; k++)
        sum++
```