

C241 Homework 5: Sorting, 'Big-O' notation, Computational Complexity

Due Wednesday 2/27/08

1) Use the bubble sort algorithm to put the list (5, 1, 4, 2, 8) into increasing order. Write down all intermediate lists and indicate the elements that have been compared. Also indicate if the compared pairs have been interchanged or not.

2) Use the merge sort algorithm to put the list (38, 27, 43, 3, 9, 82, 10) into increasing order. Write down all intermediate lists. The end-result should look like a *tree* with one root node (the given list) and one final node (the sorted list). The *tree* should get "wider" as lists are being split and "narrow down" as lists are being merged.

3) Find the asymptotic complexity (the best 'Big-O') for the following functions:

a) $f(n) = 3n + 7$

c) $f(n) = 3n^3 - 5n^2 + 25n - 165$

d) $f(n) = 5n^2 + 3n\log_2(n)$

e) $f(n) = n^2 + (n - 1)^3$

f) $f(n) = \frac{n(n+1)(n+2)}{(n+3)}$

g) $f(n) = 2 + 4 + \dots + 2n$

4) Let $f(n) = n + 100$ and $g(n) = n^2$. Use the definition of Big-O to prove that $f(n) \in O(g(n))$

5) Let $f(n) = n$ and $g(n) = n + (1/n)$. Use the definition of Big-O to prove that $f(n) \in O(g(n))$ and $g(n) \in O(f(n))$.

6) Find the time complexity of the following pieces of code:

a)

```
for(i : 1 to n)
```

```
  x = x + 1
```

b)

```
for(i : 1 to n)
```

```
  for (j:1 to m)
```

```
    x = x + 1
```

c) (note that the index k is used in both loops)

```
for (k : 1 to n)
```

```
  for (j: 1 to k)
```

```
    x = x + 1
```

d)

```
for (i : 1 to n)
```

```
  y = y + 1
```

```
  for (h: 1 to p)
```

```
    z = z + 3
```

```
  for (k: 1 to q)
```

```
    x = x + 1
```

e)

```
for (k : 1 to n)
```

```
  for (j: 1 to  $n^2$ )
```

```
    x = x + 1
```