ECS 122A Homework 1

Hardy Jones
999397426
Professor Bai
Spring 2014

1. We hope to solve $8n^2 < 64n\lg n$. We can get fairly far with just algebraic manipulations.

$$\begin{aligned} 8n^2 &< 64n\lg n \\ n^2 &< 8n\lg n \\ n &< 8\lg n \\ n &< \lg n^8 \\ 2^n &< n^8 \end{aligned}$$

   After a bit of guess and check, we see that this is valid for $1 < n < 44$.

   So Insertion sort beats merge sort for arrays of length $1 < n < 44$.

2. *Proof.* **Base Case** n = 2

$$T(2) = 2lg2 = 2$$

   Inductive Case

   Assume $T(n) = n\lg n$

   Show $T(2n) = 2n\lg 2n$

$$\begin{aligned} T(2n) &= 2T\left(\frac{2n}{2}\right) + 2n \\ &= 2T(n) + 2n \\ &= 2n\lg n + 2n \\ &= 2n(\lg n + 1) \\ &= 2n(\lg n + \lg 2) \\ &= 2n\lg 2n \end{aligned}$$

   Thus, By mathematical induction, our recurrence is satisfied. □

4. (a)

```
Selection Sort
input  : Array of length n
output: Sorted array of length n
for i ← 0 to n do
    smallest = i
    for j ← i to n do
        if array[i] < array[smallest] then
            smallest = j
        end
    end
    swap array[i] with array[smallest]
end
return sorted array
```

(b) Best case: $T(n) = n^2 + 2n + 1$

Worst case: $T(n) = 2n^2 + 2n + 1$

5. (a) With the exception of the initial element in an array, we have to perform 2 comparisons at every element of the array. One comparison for maximum, and one for minimum. For the initial element we can just assume that it is both the maximum and the minimum of the array.

So, we end up making 2 comparisons for $n - 1$ elements in the array, or $2n - 2$ comparisons

(b)

```
MinMax
input  : Array of length n
output: Array of with minimum and maximum
if length of array is 2 then
    if array[0] > array[1] then
        return [array[0], array[1]]
    else
        return [array[1], array[0]]
    end
else
    split array into left and right halves [min1, max1] ← MinMax(leftHalf) [min2,
    max2] ← MinMax(rightHalf) if min1 < min2 then
        trueMin ← min1
    else
        trueMin ← min2
    end
    if max1 > max2 then
        trueMax ← max1
    else
        trueMax ← max2
    end
    return [trueMin, trueMax]
end
```

(c)

$$T(n) = 1 \text{ if } n = 2$$
$$T(n) = 2T\left(\frac{n}{2}\right) - 2 \text{ if } n = 2^d \text{ for } d > 1$$