CS 344 Assignment 1

September 19, 2018

Name : Yang Bao
NetID : yb184
Section : 02
The classmate I discussed with : Zitian Qin

Problem 1 :

1. $f(n) = O(g(n))$
2. $f(n) = O(g(n))$
3. $f(n) = \Omega(g(n))$
4. $f(n) = \Theta(g(n))$
5. $f(n) = O(g(n))$
6. $f(n) = O(g(n))$
7. $f(n) = \Omega(g(n))$
8. $f(n) = O(g(n))$
9. $f(n) = \Theta(g(n))$
10. $f(n) = O(g(n))$

Problem 2 :
Part 1 :
Sol :

1. Base part :
   When k = 0, $\sum_{i=0}^{k} i2^i = 0$.
   When k = 0, $(k-1) \cdot 2^{k+1} + 2 = 0$.
   $\sum_{i=0}^{k} i2^i = (k-1) \cdot 2^{k+1} + 2$

2. Induction part :
   When k = k + 1, $\sum_{i=0}^{k+1} i2^i = \sum_{i=0}^{k} i2^i + (k+1)2^{k+1}$
   As when k = k, $\sum_{i=0}^{k} i2^i = (k-1) \cdot 2^{k+1} + 2$,
   $\sum_{i=0}^{k+1} i2^i = (k-1) \cdot 2^{k+1} + 2 + (k+1) \cdot 2^{k+1} = 2k \cdot 2^{k+1} + 2 = k \cdot 2^{k+2} + 2$
   When k = k + 1,$(k-1) \cdot 2^{k+1} + 2 = (k+1-1) \cdot 2^{k+1+1} + 2 = k \cdot 2^{k+2} + 2$
   $\sum_{i=0}^{k} i2^i = (k-1) \cdot 2^{k+1} + 2$ proved.

Part 2 :
Sol :
First, $\sum_{i=1}^{n} \frac{i^4}{10} \leq \sum_{i=1}^{n} \frac{n^4}{10} = \frac{n^5}{10} \leq n^5$. So $\sum_{i=1}^{n} \frac{i^4}{10} = O(n^5)$
Then to prove $\sum_{i=1}^{n} \frac{i^4}{10} = \Omega(n^5)$
Consider the last $\frac{n}{2}$ numbers:
$\sum_{i=\frac{n}{2}}^{n} \frac{i^4}{10} \geq \sum_{i=\frac{n}{2}}^{n} \frac{\frac{n}{2}^4}{10} = \frac{\frac{n}{2}^4}{10} \cdot \frac{n}{2} = \frac{n^5}{320}$
When $C = \frac{1}{320}$, $\sum_{i=\frac{n}{2}}^{n} \frac{i^4}{10} = \frac{n^5}{320} = C \cdot g(n) = \frac{n^5}{320}$
So $\sum_{i=1}^{n} \frac{i^4}{10} \geq \frac{1}{320} n^5$
$\sum_{i=1}^{n} \frac{i^4}{10} = \Omega(n^5)$
So $\sum_{i=1}^{n} \frac{i^4}{10} = \Theta(n^5)$

Part 3 :
Sol :
$\sum_{i=1}^{log_2(n)} 4^i \leq \sum_{i=1}^{log_2(n)} 4^{log_2(n)} = \frac{4^{log_2(n)}-1}{4-1} = \frac{n^2-1}{3} \leq n^2$
So $\sum_{i=1}^{log_2(n)} 4^i = O(n^2)$
Then to prove $\sum_{i=1}^{log_2(n)} 4^i = \Omega(n^2)$
$\sum_{i=1}^{log_2(n)} 4^i = \sum_{i=1}^{log_2(n)-1} 4^i + 4^{log_2(n)} = \sum_{i=1}^{log_2(n)-1} 4^i + n^2$
Since $\sum_{i=1}^{log_2(n)-1} 4^i > 0$, $\sum_{i=1}^{log_2(n)} 4^i \geq n^2$, so $\sum_{i=1}^{log_2(n)} 4^i = \Omega(n^2)$
So $\sum_{i=1}^{log_2(n)} 4^i = \Theta(n^2)$

Problem 3 :

Part 1 :

Sol :

The pseudocode for Closest Pair is :

```
closestPair(A)
    quicksort(A,0,A.length-1)     //sort the array first
    dif = |A[0] - A[1]|
    index = 0
    for i = 0 to A.length-2
        if |A[i] - A[i+1]| < dif
            dif = |A[i] - A[i+1]|
            index = i
    output A[index],A[index+1]

quicksort(A, lo, hi)
    if lo < hi
        p = partition(A, lo, hi)
        quicksort(A, lo, p - 1)
        quicksort(A, p + 1, hi)

partition(A, lo, hi)
    pivot = A[hi]
    i = lo                        //place for swapping
    for j = lo to hi - 1
        if A[j] <= pivot
            swap A[i] with A[j]
            i = i + 1
    swap A[i] with A[hi]
    return i
```

More space for Problem 3 :
Part 2 :
Sol :
The pseudocode for Remove Duplicates is :

```
removeDuplicates(A)
    quicksort(A,0,A.length-1)      //sort the array first
    output A[0]
    for i = 1 to A.length-1
        if A[i] != A[i-1]
            output A[i]

quicksort(A, lo, hi)
    if lo < hi
        p = partition(A, lo, hi)
        quicksort(A, lo, p - 1)
        quicksort(A, p + 1, hi)

partition(A, lo, hi)
    pivot = A[hi]
    i = lo                         //place for swapping
    for j = lo to hi - 1
        if A[j] <= pivot
            swap A[i] with A[j]
            i = i + 1
    swap A[i] with A[hi]
    return i
```

Problem 4 :
Sol :
Obviously the total running time for the algorithm is $\sum_{i,j>i}(j-i)$.
First, to prove that $\sum_{i,j>i}(j-i) = O(n^3)$:
$\sum_{i,j>i}(j-i) = \sum_{i=0}^{n}(\sum_{j=i}^{n}(j-i)) = \sum_{i=0}^{n}(i+(i-1)+(i-2)+\cdots+1+0)$
$\leq \sum_{i=0}^{n}(i+i+i+\cdots+i+i) = \sum_{i=0}^{n} i^2 \leq \sum_{i=0}^{n} n^2 = n^3$
So $\sum_{i=1}^{n} \frac{i(i+1)}{2} \leq n^3$, so $\sum_{i,j>i}(j-i) = O(n^3)$.

Second, to prove that $\sum_{i,j>i}(j-i) = \Omega(n^3)$:
$\sum_{i,j>i}(j-i) = \sum_{i=0}^{n}(\sum_{j=i}^{n}(j-i)) = \sum_{i=0}^{n}(i+(i-1)+(i-2)+\cdots+1+0) = \sum_{i=0}^{n} \frac{i(i+1)}{2}$
$\geq \sum_{i=0}^{n} \frac{i^2}{2}$
Consider the last $\frac{n}{2}$ elements in that polynomial we get :
$\sum_{i=\frac{n}{2}}^{n} \frac{i^2}{2} \geq \sum_{i=\frac{n}{2}}^{n} \frac{\frac{n}{2}^2}{2} = \sum_{i=\frac{n}{2}}^{n} \frac{n^2}{8} = \frac{n}{2} \cdot \frac{n^2}{8} = \frac{n^3}{16}$
So when $C = 16$, we have $C \cdot \sum_{i,j>i}(j-i) \geq n^3$, so $\sum_{i,j>i}(j-i) = \Omega(n^3)$

So $\sum_{i,j>i}(j-i) = \Theta(n^3)$