Name: shi qiu
Access ID: sbq5043
Recitation: 7

Problem 1

Points:

“I did not work in a group.”

“I did not consult anyone except my group members”.

non-class material: https://www.geeksforgeeks.org/merge-two-sorted-arrays/

Problem 2

Points:

1. $\Theta(n^3 logn)$
2. $\Theta(n^3.5)$
3. $\Theta(n^4)$
4. $\Theta(nlog^2 n)$
5. $\Theta(n^{3.5} log^3 n)$

Problem 3

Points:

1. the first time merging first two arrays will cost n+n time and total length will be 2n

the merging of third in to the first two will cost 2n+n time and total length will be 3n

so until the last, it will cost (m-1)*n+n time and total length will be mn

if we add them tougher, it will cost $O(n * m^2)$

2. if we use divided and concur which introduced in class, we split the array in to half every time it will be near linear$O(mlogm)$, and total will cost $O(n * mlogm)$

Problem 4 **Points:**

for the pivot function here, we first divided into n/3 and calculate the medians, then divided in to n/9 and return selections

total run time is $\Theta(n)+\Theta(n/3*log3)+\Theta(n)+\Theta(n/9*log9)$

for n/3:

$M=n/3$

$A1>M/2+M/2=2*M/2=M=n/3$

$A2>M/2+M/2=2*M/2=M=n/3$

$a=1/3b=2/3a+b=1$

$T(n)=\Theta(nlogn)$

for n/9:

$M=n/9$

$A1>M/2+4*M/2=5*M/2=M=5n/18$

$A2>M/2+4*M/2=5*M/2=M=5n/18$

$a=1/9b=13/18a+b<1$

$T(n)=\Theta(n)$

and use master theorem to have $T(n)=\Theta(n^3logn)$

Problem 5

Points:

no idea