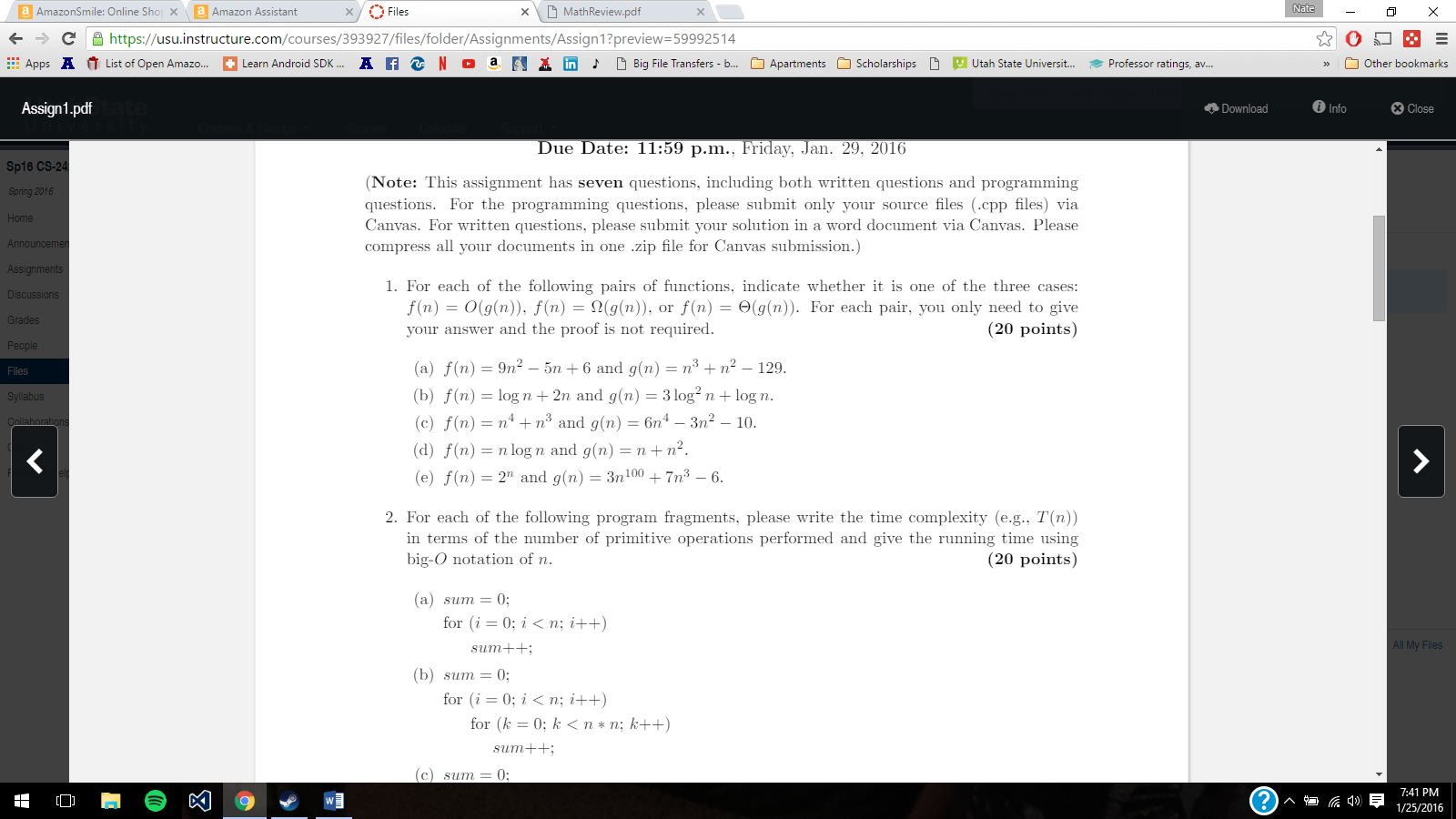
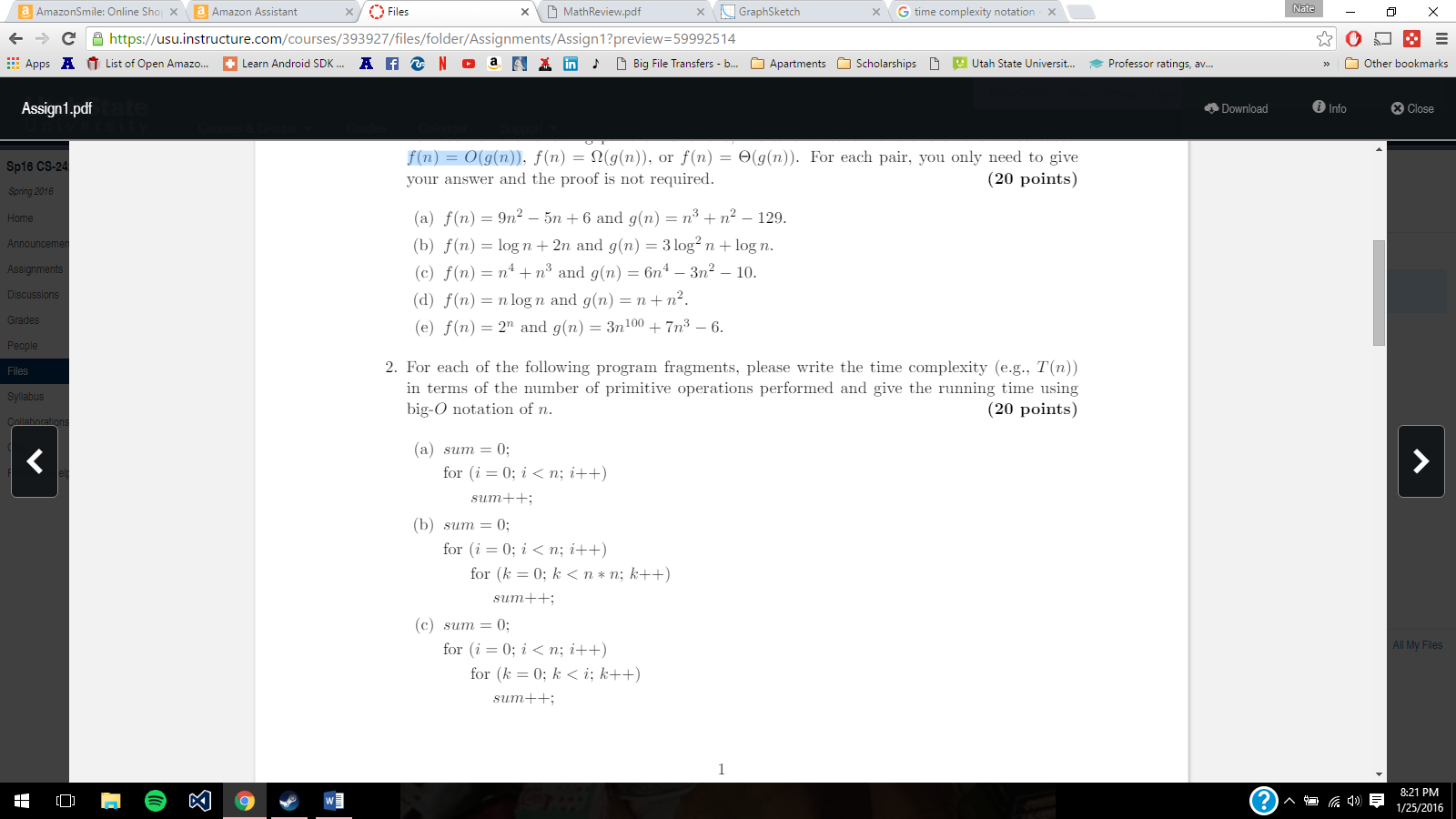
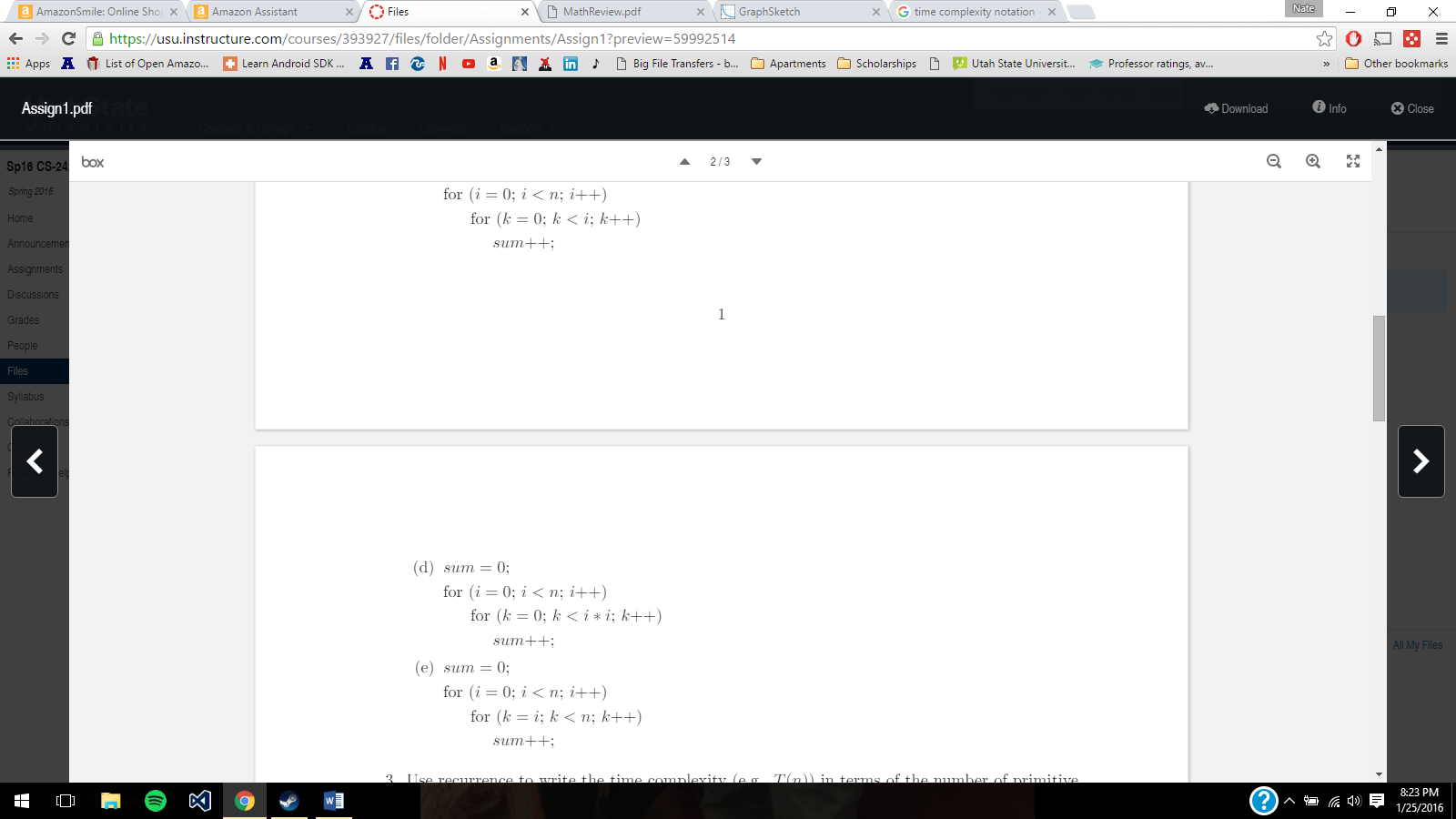
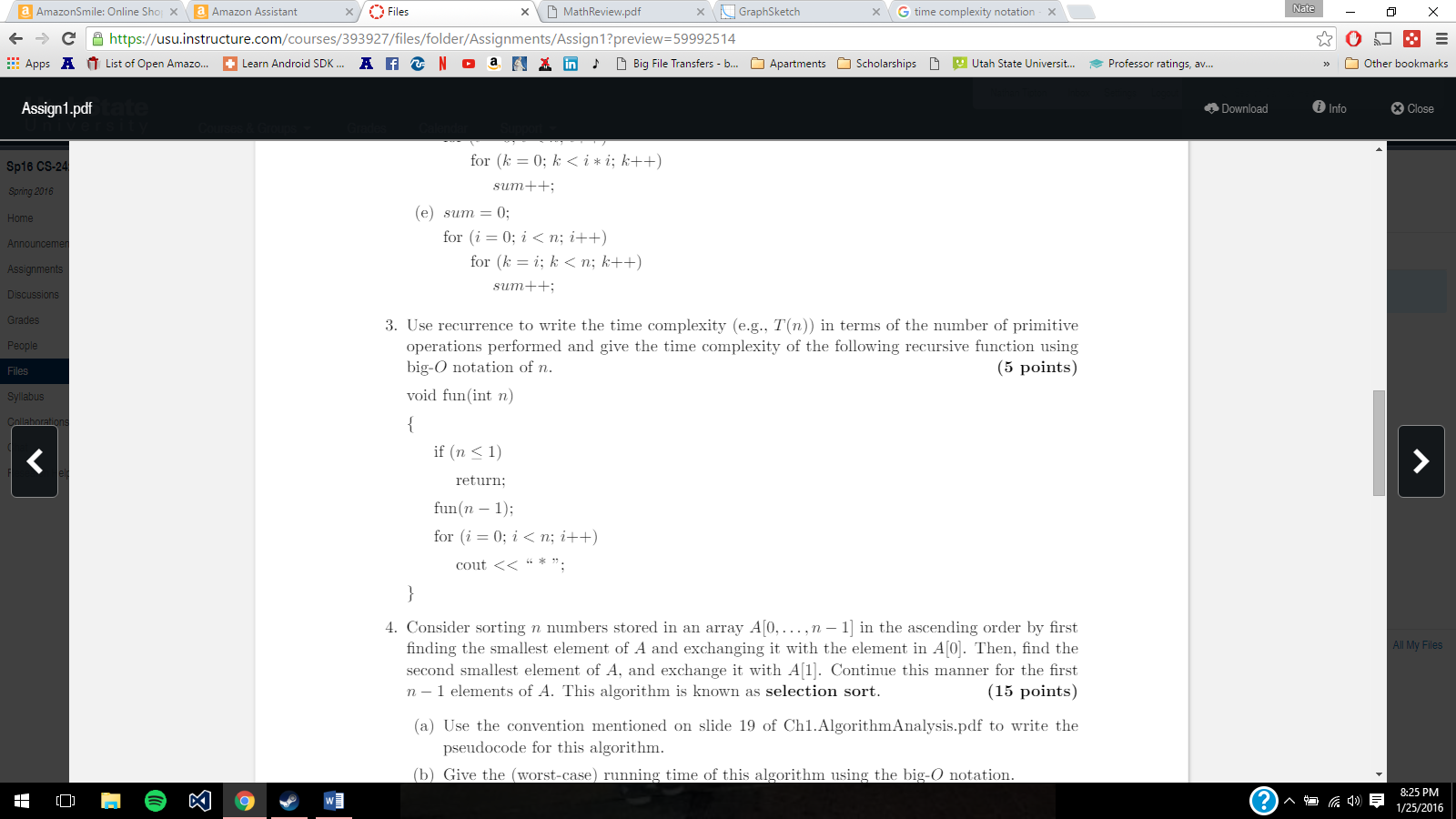
Nathan Tipton A01207112 Assignment 1 – CS 2420



1. f(n) = O(n^3)
2. f(n)=Ω(log n)
3. f(n) = Θ(n^4)
4. f(n) = O(n^2)
5. f(n) = O(n^100)



1. T(n)=(n+n)+1=2n+1=O(n)
2. T(n) =1+n(2n^2)=O(n^3)
3. T(n) ≤(n+1)(n/2)=1+n^2/2=O(n^2)
4. T(n)=1+n(n/2)^2=1+n^3/4=O(n^3)
5. T(n)=1+n^2/2=O(n^2)

3. T(n)=T(n-1)+n T(n-1)=T(n-1-1)+(n-1)=T(n-2)+(n-1)

T(n)=T(n-2)+(n-1)+n

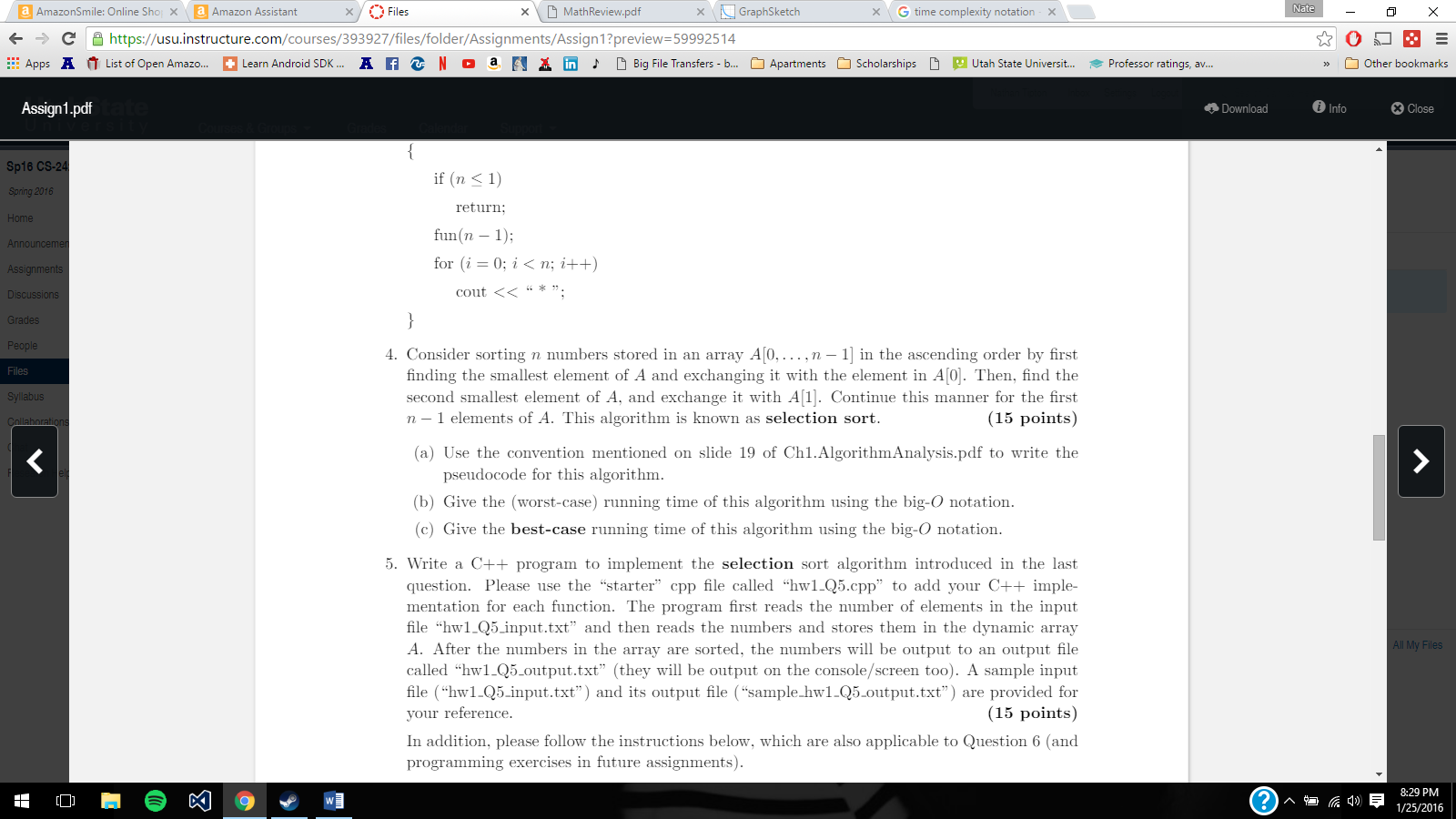
n+(n-1)+(n-2)+(n-3)+………..2+1

T(n)=n(n+1)/2= (n^2/2)+(n/2)

dropping lower order =n^2/2

dropping constant = n^2

Therefore T(n)<=n(n+1)/2=O(n^2)

A. for i←0 to size{

min←i

for j←i+1 to size{

if A[j]<A[min] then min←j

}

if min !=i {

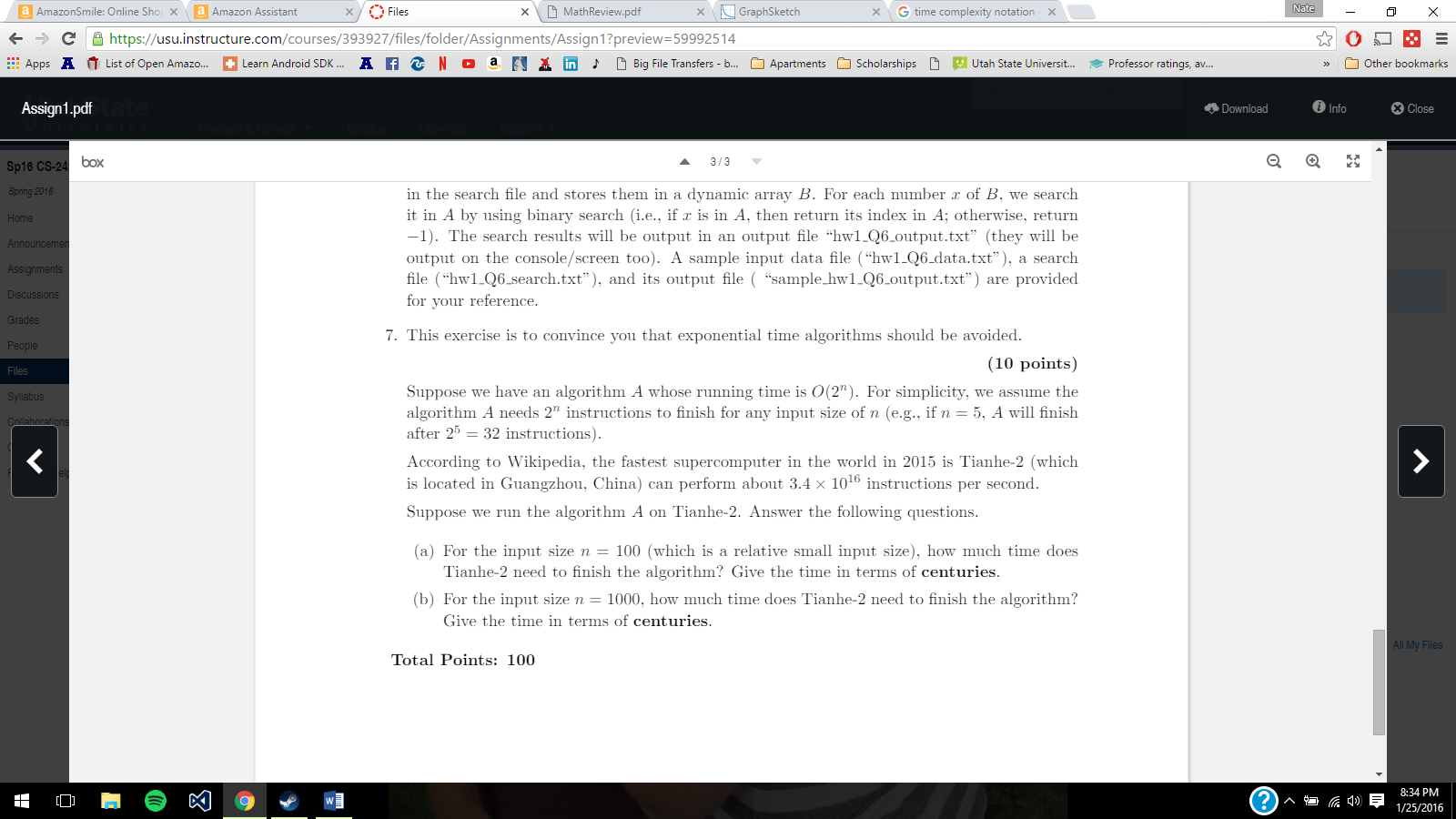
then temp←A[i]

A[i]=A[min]

A[i]=temp }

B.O(n^2), all elements are unsorted, has to swap all elements, go through the entire array comparting and swapping

C.O(n^2), elements are already sorted, however algorithm still has to run through the entire array comparing element values



A. 2^100 1.2676506\*10^30 instructions. 3.4\*10^16 instructions per second.

=3.7283841\*10^13 seconds

=621397353053 minutes

=10356622550 hours

=431525939 days

=1182262.848 years

=11822.63 centuries

B. =9.986487894\*10^274 Centuries