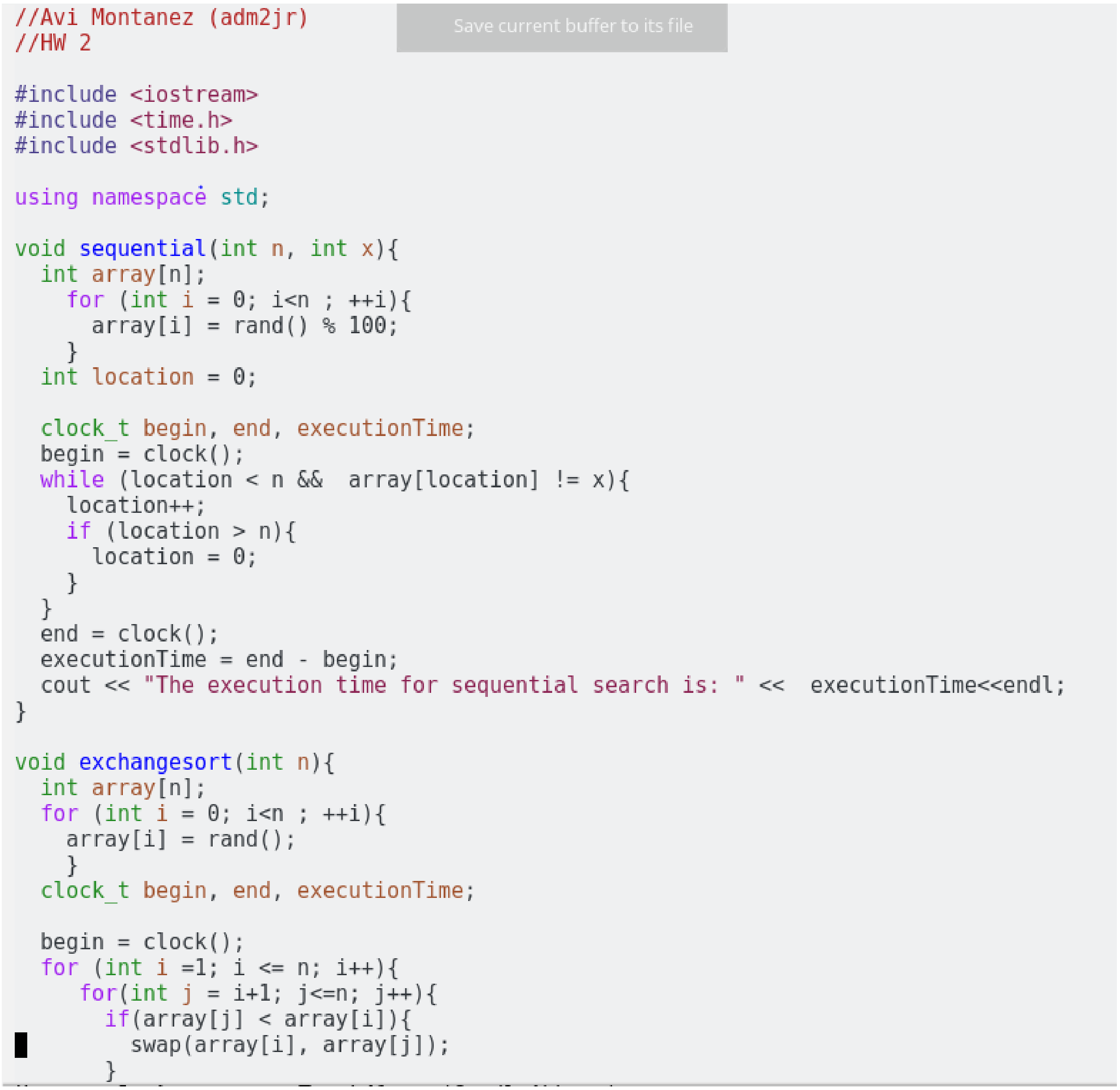
Avi Montanez (adm2jr)

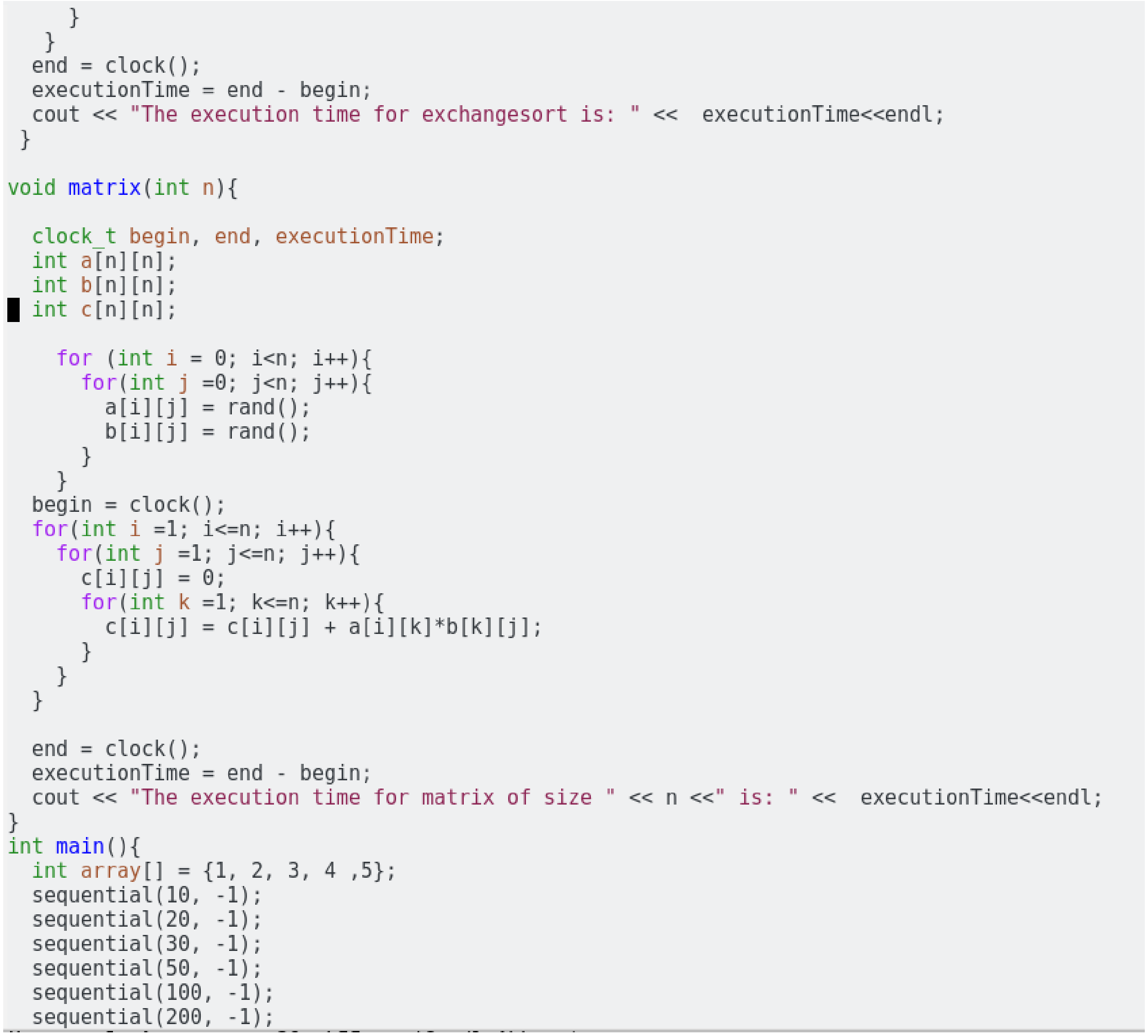
2/3/2017

CS 4102 HW2

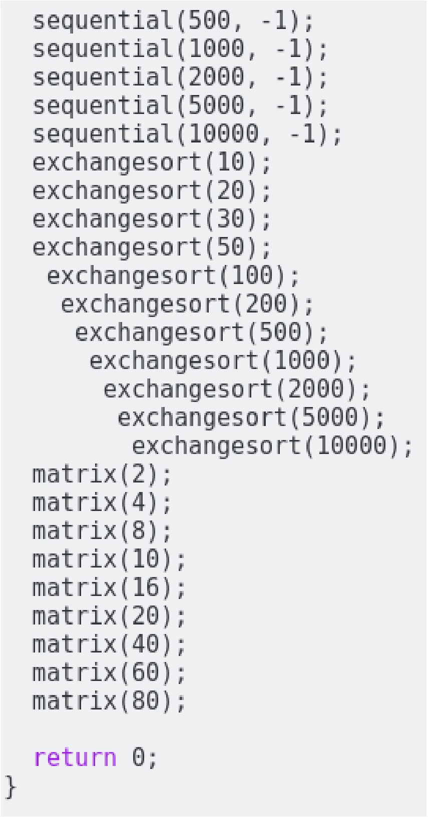
On my honor, I have neither given nor received aid on this assignment



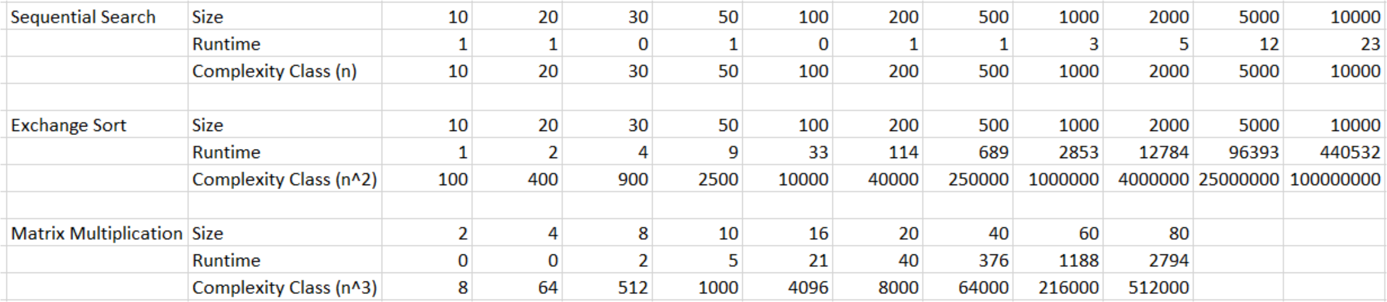
```
//Avi Montanez (adm2jr)
//HW 2

#include <iostream>
#include <time.h>
#include <stdlib.h>

using namespace std;

void sequential(int n, int x){
  int array[n];
    for (int i = 0; i<n ; ++i){
      array[i] = rand() % 100;
    }
  int location = 0;

  clock_t begin, end, executionTime;
  begin = clock();
  while (location < n &&  array[location] != x){
    location++;
    if (location > n){
      location = 0;
    }
  }
  end = clock();
  executionTime = end - begin;
  cout << "The execution time for sequential search is: " <<  executionTime<<endl;
}

void exchangesort(int n){
  int array[n];
  for (int i = 0; i<n ; ++i){
    array[i] = rand();
    }
  clock_t begin, end, executionTime;

  begin = clock();
  for (int i =1; i <= n; i++){
     for(int j = i+1; j<=n; j++){
       if(array[j] < array[i]){
         swap(array[i], array[j]);
       }
```



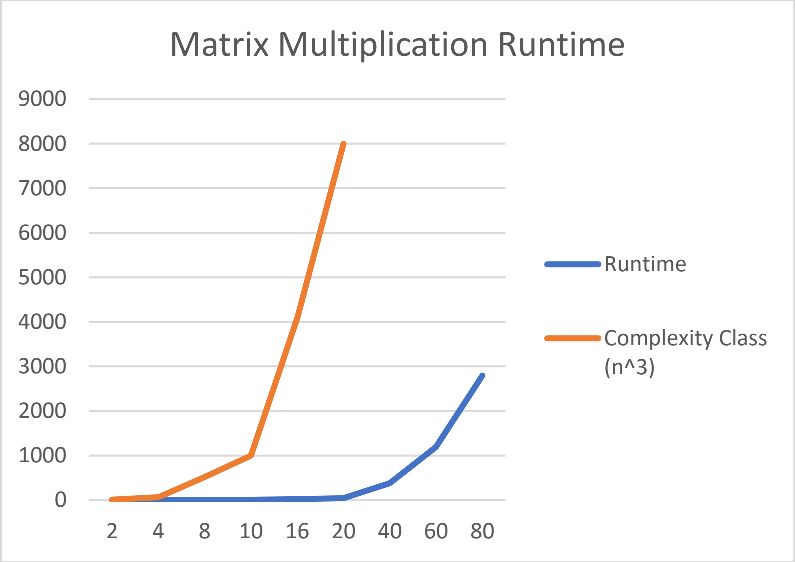
```
     }
   }
  end = clock();
  executionTime = end - begin;
  cout << "The execution time for exchangesort is: " <<  executionTime<<endl;
 }

void matrix(int n){

  clock_t begin, end, executionTime;
  int a[n][n];
  int b[n][n];
  int c[n][n];

    for (int i = 0; i<n; i++){
      for(int j =0; j<n; j++){
        a[i][j] = rand();
        b[i][j] = rand();
      }
    }
  begin = clock();
  for(int i =1; i<=n; i++){
    for(int j =1; j<=n; j++){
      c[i][j] = 0;
      for(int k =1; k<=n; k++){
        c[i][j] = c[i][j] + a[i][k]*b[k][j];
      }
    }
  }

  end = clock();
  executionTime = end - begin;
  cout << "The execution time for matrix of size " << n <<" is: " <<  executionTime<<endl;
}
int main(){
  int array[] = {1, 2, 3, 4 ,5};
  sequential(10, -1);
  sequential(20, -1);
  sequential(30, -1);
  sequential(50, -1);
  sequential(100, -1);
  sequential(200, -1);
```

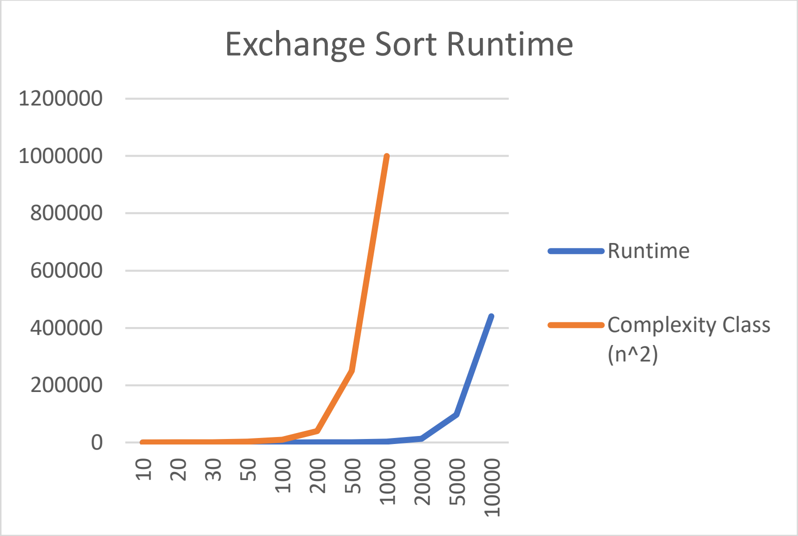


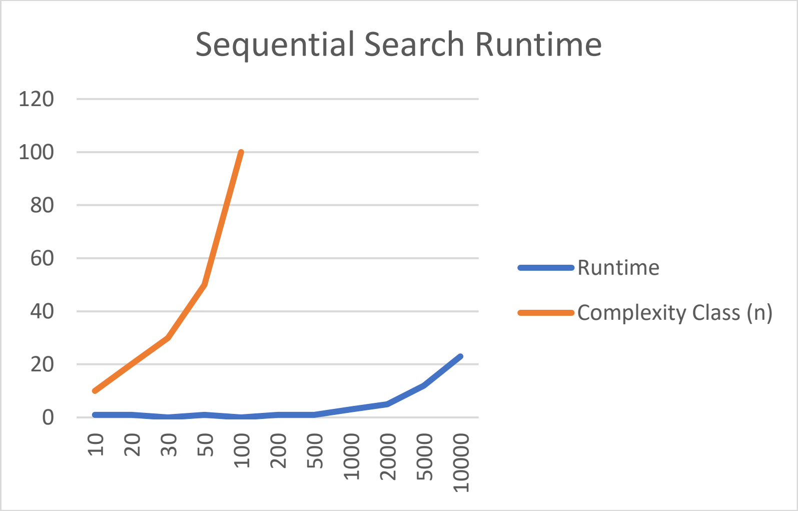
```
  sequential(500, -1);
  sequential(1000, -1);
  sequential(2000, -1);
  sequential(5000, -1);
  sequential(10000, -1);
  exchangesort(10);
  exchangesort(20);
  exchangesort(30);
  exchangesort(50);
   exchangesort(100);
    exchangesort(200);
     exchangesort(500);
      exchangesort(1000);
       exchangesort(2000);
        exchangesort(5000);
         exchangesort(10000);
  matrix(2);
  matrix(4);
  matrix(8);
  matrix(10);
  matrix(16);
  matrix(20);
  matrix(40);
  matrix(60);
  matrix(80);

  return 0;
}
```



Sequential Search	Size	10	20	30	50	100	200	500	1000	2000	5000	10000
	Runtime	1	1	0	1	0	1	1	3	5	12	23
	Complexity Class (n)	10	20	30	50	100	200	500	1000	2000	5000	10000
Exchange Sort	Size	10	20	30	50	100	200	500	1000	2000	5000	10000
	Runtime	1	2	4	9	33	114	689	2853	12784	96393	440532
	Complexity Class (n^2)	100	400	900	2500	10000	40000	250000	1000000	4000000	25000000	100000000
Matrix Multiplication	Size	2	4	8	10	16	20	40	60	80		
	Runtime	0	0	2	5	21	40	376	1188	2794		
	Complexity Class (n^3)	8	64	512	1000	4096	8000	64000	216000	512000		


Matrix Multiplication Runtime
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
2
4
8
10
16
20
40
60
80
Runtime
Complexity Class (n^3)


Exchange Sort Runtime
1200000
1000000
800000
600000
400000
200000
0
10
20
30
50
100
200
500
1000
2000
5000
10000
Runtime
Complexity Class (n^2)


Sequential Search Runtime
120
100
80
60
40
20
0
10
20
30
50
100
200
500
1000
2000
5000
10000
Runtime
Complexity Class (n)

For every single function, it ran much faster than its respective complexity class. I believe this is due to the nature of the length in time of a clock cycle. This may be due to how many computations can be done within the length of one clock tick. Maybe the function behavior would have been different and more consistent with what we learned in class if runtime was measured in seconds or another time unit.