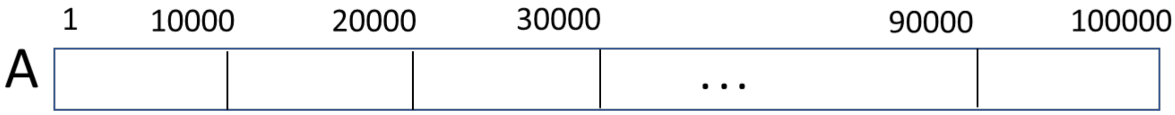
PROGRAMMING PROJECT, more information

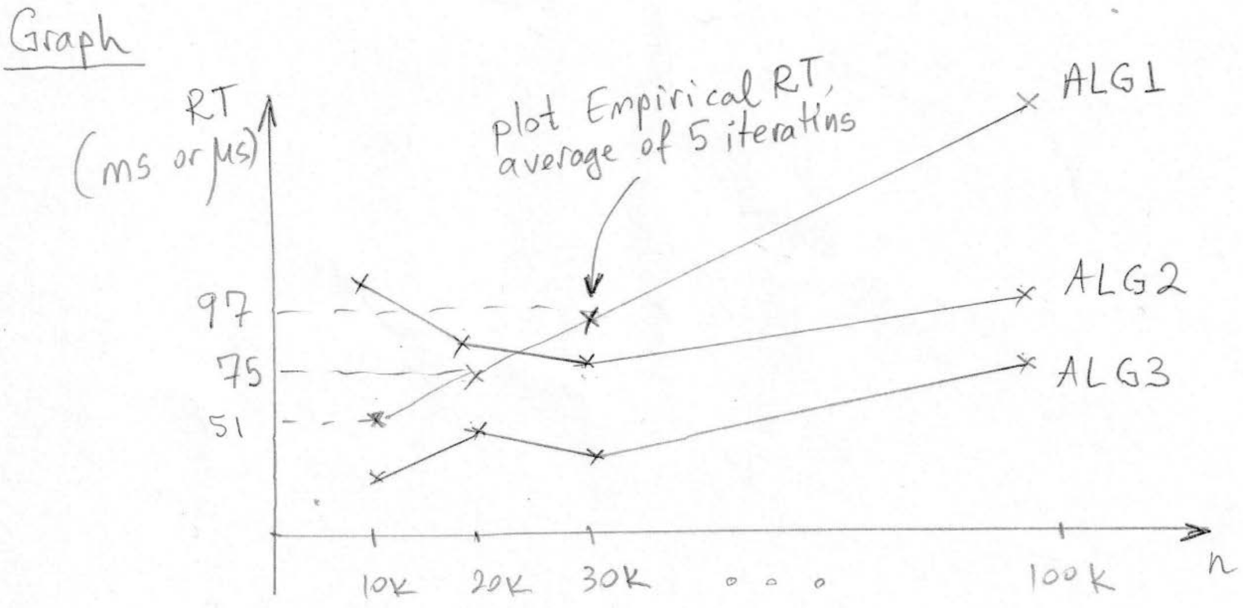
- Start by implementing the 3 algorithms ALG1, ALG2, and ALG3 and their supporting functions, and test on small values of n to test for correctness. Important: follow the pseudocode from the textbook/notes.



```
main {
max = 100000
m = 5

for k = 1 to m    // for each iteration k
      for  j = 1 to max
              A[j] = rand()
      for n = 10000; n <= 100000; n = n + 10000
              i = $\lfloor 2n/3 \rfloor$
            // measurements for ALG1
              B[1…n] = A[1…n]
              t1 = time()
              ALG1 (B, n , i)
              t2 = time()
              $t_{ALG1}$ [k,n] =  t2 – t1
              // measurements for ALG2
              B[1…n] = A[1…n]
              t1 = time()
              ALG2 (B, n , i)
              t2 = time()
              $t_{ALG2}$ [k,n] =  t2 – t1
              // measurements for ALG3
              B[1…n] = A[1…n]
              t1 = time()
              ALG3 (B, n , i)
              t2 = time()
              $t_{ALG3}$ [k,n] =  t2 – t1
//compute the average values
for  n = 10000; n <= 100000; n = n + 10000
       t_AVG_ALG1[n] = ( $t_{ALG1}$ [1,n] + $t_{ALG1}$ [2,n] + ……+ $t_{ALG1}$ [m,n] )/m
        t_AVG_ALG2[n] = ( $t_{ALG2}$ [1,n] + $t_{ALG2}$ [2,n] + ……+ $t_{ALG2}$ [m,n] )/m
        t_AVG_ALG3[n] = ( $t_{ALG3}$ [1,n] + $t_{ALG3}$ [2,n] + ……+ $t_{ALG3}$ [m,n] )/m
}
```

- Note: when you allocate the array A, feel free to use dynamic allocation using pointers for C, vector for C++, etc.
- Note that t_AVG_ALG1, t_AVG_ALG2, t_AVG_ALG3 are the values that you plot in the graph (i.e. EmpiricalRT), and the values for the tables for the EmpiricalRT


Graph
RT
(ms or μs)
plot Empirical RT,
average of 5 iterations
ALG1
ALG2
ALG3
10K
20K
30K
° ° °
100k
n