Pseudocode

enum dataState:

presorted,

reversesorted,

random

struct record:

size,

state,

mergeTime, bubbleTime, insertionTime

RunAlgs( size, dataState ):

GenerateData(B, size, dataState)

A=B

startTime = currentTime()

mergesort(A, 0, size-1) // we index from 0

mergesortTime = currentTime() – startTime

A=B

startTime = currentTime()

bubblesort(A, 0, size-1) // we index from 0

bubblesortTime = currentTime() – startTime

A=B

startTime = currentTime()

insertionsort(A, 0, size-1) // we index from 0

insertionsortTime = currentTime() – startTime

return record( size, dataState, mergesortTime, bubblesortTime, insertionsortTime )

main():

times = []

// Let’s test powers of sqrt(2),

// by doing so we will see how powers of 2 affect mergesort.

// Loop invariant: at each comparison of the guard,

// times contains data from k-1 tests of data sizes of

// the floors of the first k-1 powers of sqrt(2).

for( long k = 1; k < MAX\_EXPONENT; k++ ):

times.append( RunAlgs( floor(sqrt(2) ^ k), random ) )

times.append( RunAlgs( floor(sqrt(2) ^ k), presorted ) )

times.append( RunAlgs( floor(sqrt(2) ^ k), reversesorted ) )

output\_to\_file( times )