

SortTimes

Samyak Ahuja

August 3, 2018

Complexity for different Sorting Algorithms.

Helper Functions

Replicator

```
replicator <- function(func, size = 1000){  
  if(size == 1000){  
    ele <- seq(from = 0, to = 1000, by = 50)  
  }else{  
    ele <- seq(from = 0, to = 10000, by = 250)  
  }  
  ele <- ele[-1]  
  timeElapsed <- c()  
  for(n in ele){  
    op <- 0  
    for(i in 1:10){  
      op = op + func(sample(x = 1:100, size = n, replace = TRUE))$operations  
    }  
    op = op / 10  
    timeElapsed <- c(timeElapsed, op)  
  }  
  return (data.frame(ele,timeElapsed))  
}
```

Plotter

```
plotter <- function(df, df_title){  
  ggplot(df, aes(ele, timeElapsed, color = timeElapsed)) +  
    geom_point(shape = 16, size = 5, show.legend = FALSE, alpha = 0.6) +  
    stat_smooth(method="loess", formula=y~x) +  
    theme_minimal() +  
    labs(subtitle = "Time vs Size",  
         y = "Number of Elementary Operations (Averaged)",  
         x = "Number of Elements",  
         title = df_title) +  
    scale_color_gradient(low = "#32aeff", high = "#f2aeff") +  
    stat_poly_eq(parse=T, aes(label = ..eq.label..), formula=y~x)  
}
```

Combined Plotter

```
comb_plotter <- function(df, df_title){  
  ggplot(df, aes(ele, value, col = variable)) +  
  geom_point(shape = 16, size = 2, alpha = 0.6) +  
  stat_smooth(method="loess", formula=y~x) +  
  theme_minimal() +  
  labs(subtitle = "Time vs Size",  
       y = "Number of Elementary Operations",  
       x = "Number of Elements",  
       title = df_title) +  
  stat_poly_eq(parse=T, aes(label = ..eq.label..), formula=y~x)  
}
```

Insertion Sort

Sorting Algorithm

```
insertionSort <- function(vec){  
  n <- length(vec)  
  op <- 1  
  for(i in 2:n){  
    key <- vec[i]  
    pos <- i - 1  
    op <- op + 3  
    while(pos > 0 && vec[pos] > key){  
      vec[pos + 1] = vec[pos]  
      pos = pos - 1  
      op <- op + 4  
    }  
    vec[pos + 1] <- key  
    op <- op + 1  
  }  
  return (list("vec" = vec, "operations" = op))  
}
```

Proof of concept

```
cat(insertionSort(c(1,2,99,-21,2,23,1))$vec, "\n")
```

```
## -21 1 1 2 2 23 99
```

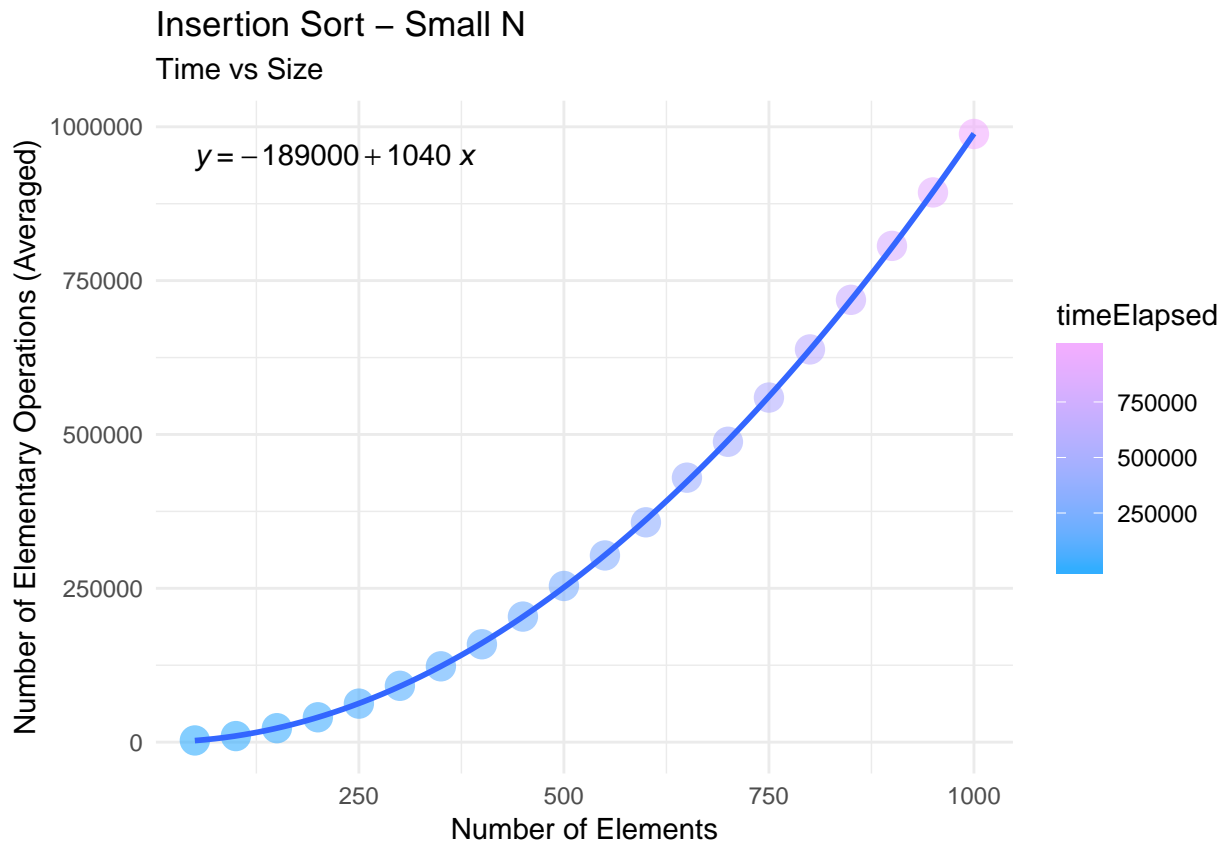
RunTime and Plot

```
isdf_small <- replicator(insertionSort)  
isdf_small
```

```
##      ele timeElapsed  
## 1      50      2749.0  
## 2     100     9862.2
```

```
## 3 150 22693.4
## 4 200 40433.8
## 5 250 62607.4
## 6 300 91558.2
## 7 350 123224.2
## 8 400 159070.2
## 9 450 204071.0
## 10 500 253811.4
## 11 550 303413.0
## 12 600 357255.8
## 13 650 429787.8
## 14 700 488141.8
## 15 750 559999.4
## 16 800 638215.8
## 17 850 718796.2
## 18 900 806459.4
## 19 950 893316.6
## 20 1000 988319.4
```

```
plotter(isdf_small, "Insertion Sort - Small N")
```



```
isdf_big <- replicator(insertionSort, 10000)
isdf_big
```

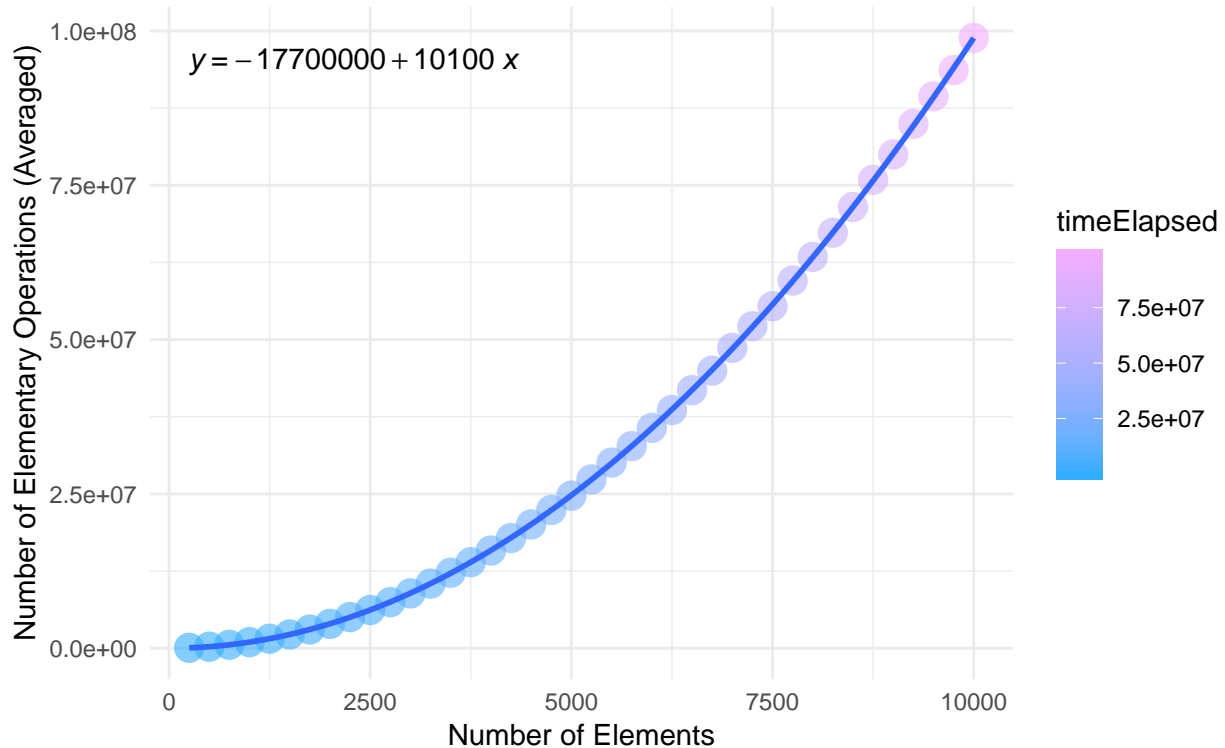
```
##     ele timeElapsed
## 1  250    63201.8
## 2  500   247054.2
## 3  750   562903.4
```

```
## 4    1000    982349.4
## 5    1250   1536933.4
## 6    1500   2234135.0
## 7    1750   3021561.0
## 8    2000   3942362.2
## 9    2250   5041619.4
## 10   2500   6191853.0
## 11   2750   7459367.4
## 12   3000   8883405.8
## 13   3250  10467540.2
## 14   3500  12226580.2
## 15   3750  13932106.2
## 16   4000  15824639.4
## 17   4250  17852322.2
## 18   4500  20055156.2
## 19   4750  22398260.6
## 20   5000  24713199.8
## 21   5250  27347217.8
## 22   5500  30070327.0
## 23   5750  32735911.0
## 24   6000  35687765.8
## 25   6250  38588668.2
## 26   6500  41808637.0
## 27   6750  44961291.4
## 28   7000  48656159.0
## 29   7250  52160421.0
## 30   7500  55393425.8
## 31   7750  59567841.4
## 32   8000  63386923.4
## 33   8250  67302632.2
## 34   8500  71475986.6
## 35   8750  75883625.4
## 36   9000  79980967.4
## 37   9250  84931350.2
## 38   9500  89409865.4
## 39   9750  93656091.0
## 40  10000  98880389.0
```

```
plotter(isdf_big, "Insertion Sort - Large N")
```

Insertion Sort – Large N

Time vs Size



Merge Sort

Sorting Algorithm

```
mergeSort <- function(vec){  
  
  mergeTwo <- function(left,right){  
    op <- 0  
    res <- c()  
    while(length(left) > 0 && length(right) > 0){  
      op <- op + 5  
      if(left[1] <= right[1]){  
        res <- c(res,left[1])  
        left <- left[-1]  
      }else{  
        res <- c(res,right[1])  
        right <- right[-1]  
      }  
    }  
  }  
  if(length(left) > 0){  
    res <- c(res,left)  
    op <- op + length(left)  
  }  
  if(length(right) > 0){  
    res <- c(res,right)
```

```

    op <- op + length(right)
  }
  op <- op + 2
  return (list("vec" = res, "operations" = op))
}

op <- 0
n <- length(vec)
if(n <= 1) return (list("vec" = vec, "operations" = op + 1))
else{
  op <- op + 1 # 1 added for previous if
  middle <- length(vec) %/% 2 #integer division
  left_list <- mergeSort(vec[1:middle])
  right_list <- mergeSort(vec[(middle + 1):n])
  left <- left_list$vec
  right <- right_list$vec
  res <- mergeTwo(left, right)
  op <- op + left_list$operations + right_list$operations + res$operations + 4
  return (list("vec" = res$vec, "operations" = op))
}
}

```

Proof of Concept

```
mergeSort(c(12,-22,13,2,-33,2))
```

```
## $vec
## [1] -33 -22  2  2 12 13
##
## $operations
## [1] 97
```

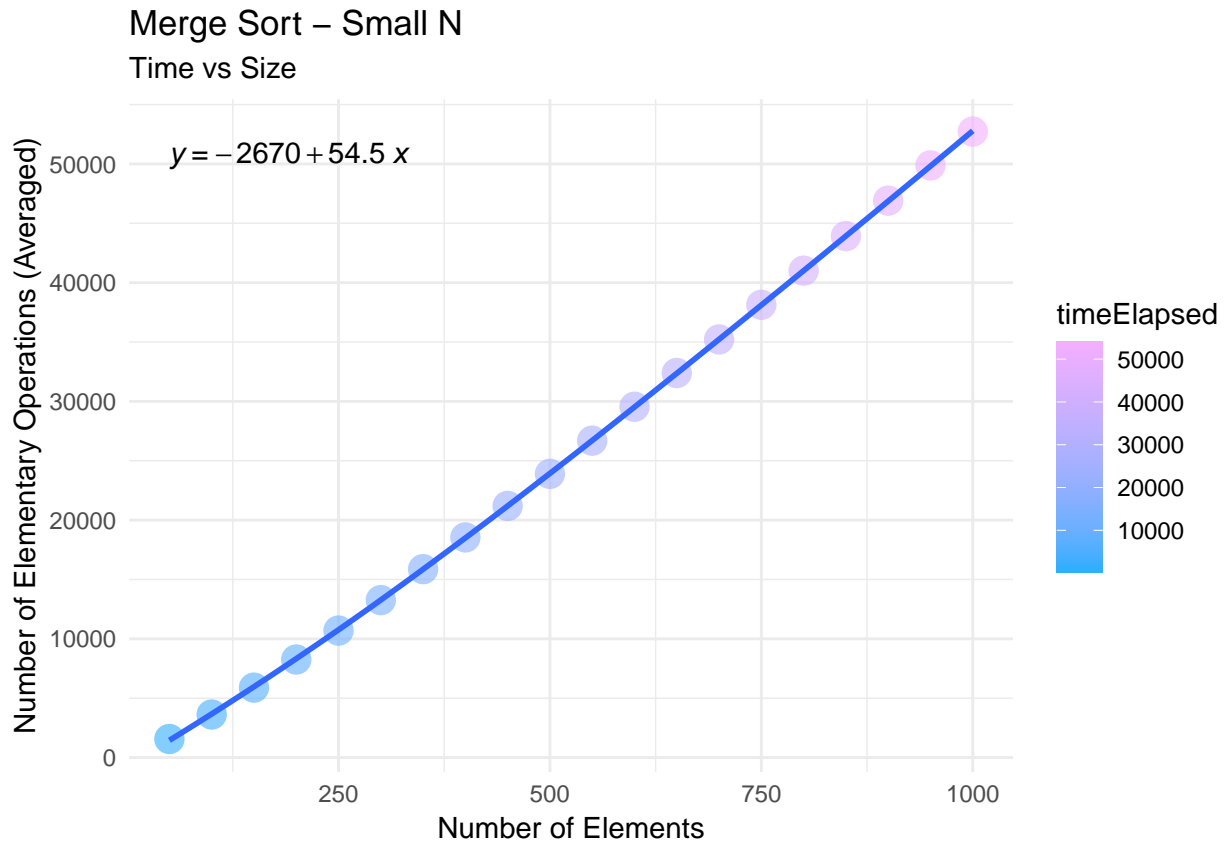
RunTime and Plot

```
msdf_small <- replicator(mergeSort)
msdf_small
```

```
##      ele timeElapsed
## 1      50      1564.2
## 2     100     3628.6
## 3     150     5890.2
## 4     200     8258.6
## 5     250    10696.6
## 6     300    13261.4
## 7     350    15869.4
## 8     400    18541.4
## 9     450    21196.6
## 10    500    23895.8
## 11    550    26689.0
## 12    600    29548.6
## 13    650    32391.0
```

```
## 14 700      35219.0
## 15 750      38136.6
## 16 800      41011.0
## 17 850      43920.6
## 18 900      46908.6
## 19 950      49874.6
## 20 1000     52727.8
```

```
plotter(msdf_small, "Merge Sort - Small N")
```



```
msdf_big <- replicator(mergeSort, 10000)
msdf_big
```

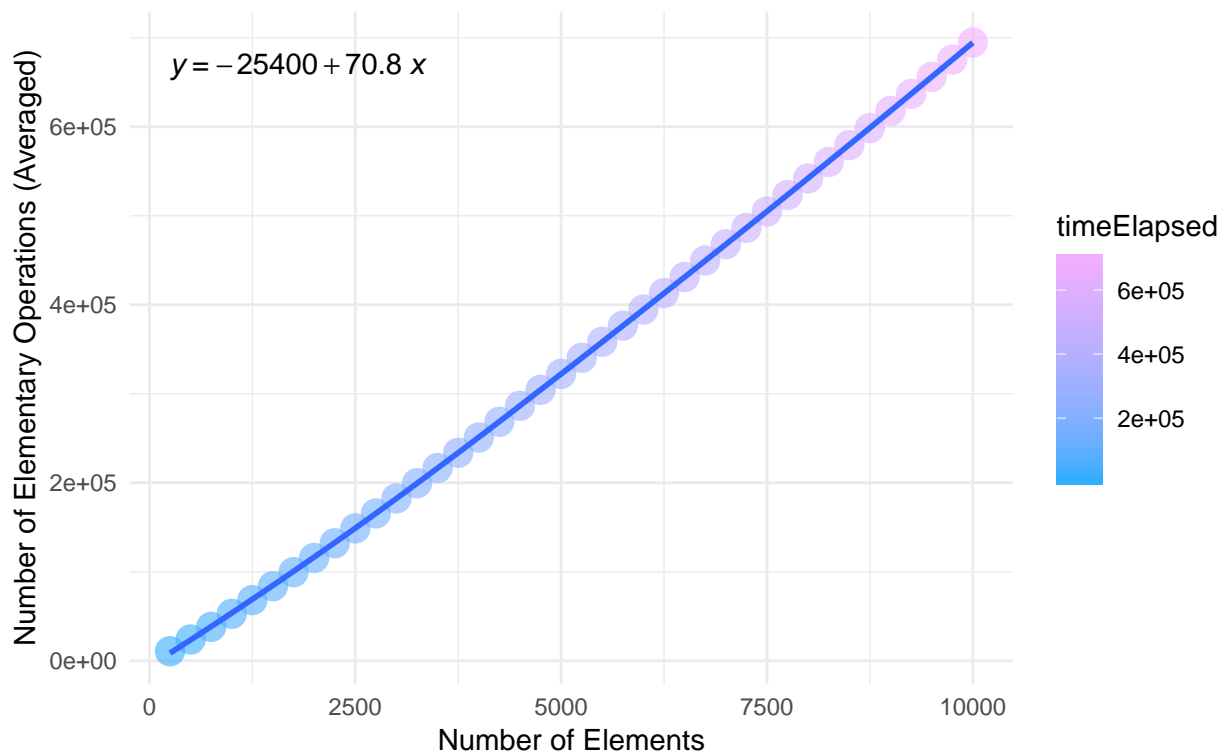
```
##      ele timeElapsed
## 1    250    10690.6
## 2    500    23881.0
## 3    750    38105.8
## 4   1000    52755.8
## 5   1250    68133.0
## 6   1500    83752.6
## 7   1750    99535.8
## 8   2000   115491.0
## 9   2250   132025.4
## 10  2500   148711.0
## 11  2750   165522.2
## 12  3000   182385.8
## 13  3250   199491.0
## 14  3500   216529.4
```

```
## 15 3750 233678.2
## 16 4000 250769.4
## 17 4250 268493.4
## 18 4500 286473.0
## 19 4750 304393.8
## 20 5000 322325.8
## 21 5250 340319.8
## 22 5500 358393.0
## 23 5750 376500.2
## 24 6000 394555.8
## 25 6250 413005.0
## 26 6500 431148.6
## 27 6750 449582.2
## 28 7000 467959.4
## 29 7250 486335.4
## 30 7500 504755.0
## 31 7750 523261.8
## 32 8000 541760.6
## 33 8250 560334.2
## 34 8500 579356.2
## 35 8750 598517.4
## 36 9000 617575.4
## 37 9250 636857.0
## 38 9500 655948.2
## 39 9750 675272.6
## 40 10000 694520.2
```

```
plotter(msdf_big, "Merge Sort - Large N")
```


Merge Sort – Large N

Time vs Size



Quick Sort

Sorting Algorithm

```
quickSort <- function(vec){
  op <- 0
  if(length(vec) > 1){
    pivot <- vec[1]
    left_list <- quickSort(vec[vec < pivot])
    right_list <- quickSort(vec[vec > pivot])
    op <- op + 4 + length(vec)
    vec <- c(left_list$vec, vec[vec == pivot], right_list$vec)
    op <- op + left_list$operations + right_list$operations + 2 + length(vec)
    return (list("vec" = vec, "operations" = op))
  }else{
    return (list("vec" = vec, "operations" = op + 1))
  }
}
```

Proof of Concept

```
quickSort(c(12,-22,13,2,-33,2))
```

```
## $vec
```

```
## [1] -33 -22  2  2 12 13
##
## $operations
## [1] 46
```

```
quickSort(c(1,1,-2,2,3,-3))
```

```
## $vec
## [1] -3 -2  1  1  2  3
##
## $operations
## [1] 42
```

```
quickSort(c(-10,-9,10,12))
```

```
## $vec
## [1] -10 -9 10 12
##
## $operations
## [1] 40
```

RunTime and Plot

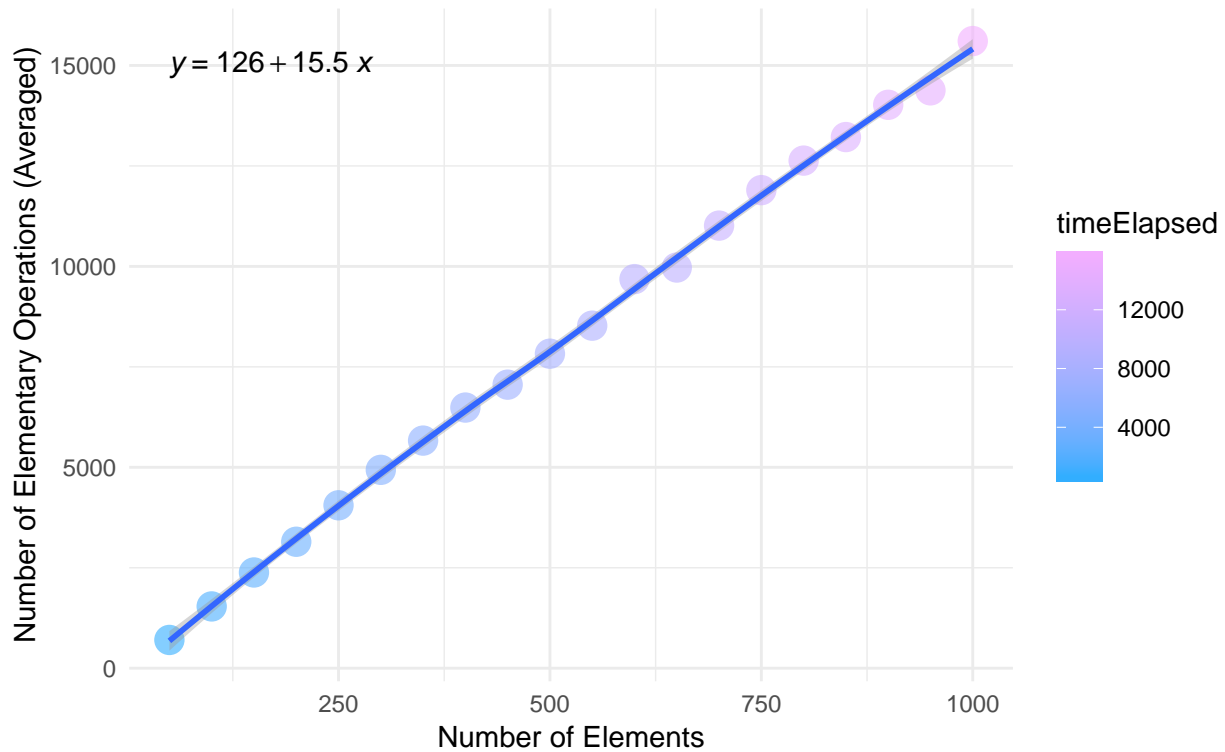
```
qsdf_small <- replicator(quickSort)
qsdf_small
```

```
##      ele timeElapsed
## 1      50         703.5
## 2     100        1537.8
## 3     150        2382.7
## 4     200        3147.3
## 5     250        4054.6
## 6     300        4936.7
## 7     350        5664.3
## 8     400        6486.4
## 9     450        7056.9
## 10    500        7827.0
## 11    550        8524.9
## 12    600        9683.3
## 13    650       9971.6
## 14    700      11014.5
## 15    750      11897.4
## 16    800      12631.7
## 17    850      13218.1
## 18    900      14020.8
## 19    950      14380.5
## 20   1000      15607.2
```

```
plotter(qsdf_small, "Quick Sort - Small N")
```

Quick Sort – Small N

Time vs Size

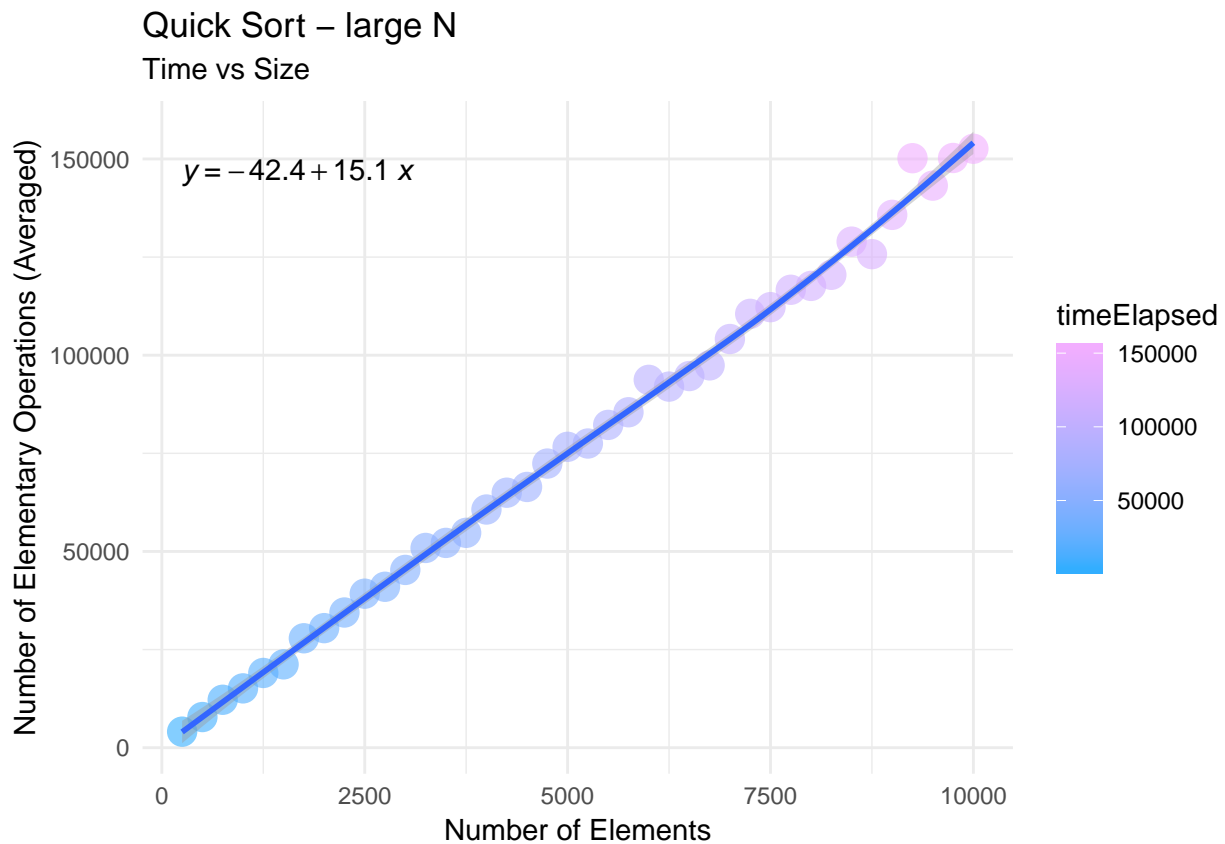


```
qsdf_big <- replicator(quickSort, 10000)
qsdf_big
```

##	ele	timeElapsed
## 1	250	4087.4
## 2	500	7812.5
## 3	750	12212.7
## 4	1000	15114.9
## 5	1250	19045.2
## 6	1500	21233.4
## 7	1750	27884.4
## 8	2000	30466.8
## 9	2250	34444.4
## 10	2500	39244.8
## 11	2750	41003.6
## 12	3000	45298.4
## 13	3250	50898.0
## 14	3500	52182.0
## 15	3750	54750.8
## 16	4000	60689.0
## 17	4250	64967.8
## 18	4500	66432.4
## 19	4750	72392.4
## 20	5000	76663.8
## 21	5250	77500.8
## 22	5500	82246.4
## 23	5750	85467.8

```
## 24 6000 93725.0
## 25 6250 92000.6
## 26 6500 94690.4
## 27 6750 97414.2
## 28 7000 104127.6
## 29 7250 110537.4
## 30 7500 112245.2
## 31 7750 116673.0
## 32 8000 117661.6
## 33 8250 120484.8
## 34 8500 128940.6
## 35 8750 125752.2
## 36 9000 135759.2
## 37 9250 150196.8
## 38 9500 143204.2
## 39 9750 150253.4
## 40 10000 152601.2
```

```
plotter(qsdf_big, "Quick Sort - large N")
```



Combined Plots

Small N

```
df_small <- data.frame(ele = msdf_small[[1]],
  insertionSort = isdf_small[[2]],
```

```

mergeSort = msdf_small[[2]],
quickSort = qsdf_small[[2]])
df_small

```

```

##      ele insertionSort mergeSort quickSort
## 1    50      2749.0    1564.2    703.5
## 2   100      9862.2    3628.6   1537.8
## 3   150     22693.4    5890.2   2382.7
## 4   200     40433.8    8258.6   3147.3
## 5   250     62607.4   10696.6   4054.6
## 6   300     91558.2   13261.4   4936.7
## 7   350    123224.2   15869.4   5664.3
## 8   400    159070.2   18541.4   6486.4
## 9   450    204071.0   21196.6   7056.9
## 10  500    253811.4   23895.8   7827.0
## 11  550    303413.0   26689.0   8524.9
## 12  600    357255.8   29548.6   9683.3
## 13  650    429787.8   32391.0   9971.6
## 14  700    488141.8   35219.0  11014.5
## 15  750    559999.4   38136.6  11897.4
## 16  800    638215.8   41011.0  12631.7
## 17  850    718796.2   43920.6  13218.1
## 18  900    806459.4   46908.6  14020.8
## 19  950    893316.6   49874.6  14380.5
## 20 1000    988319.4   52727.8  15607.2

```

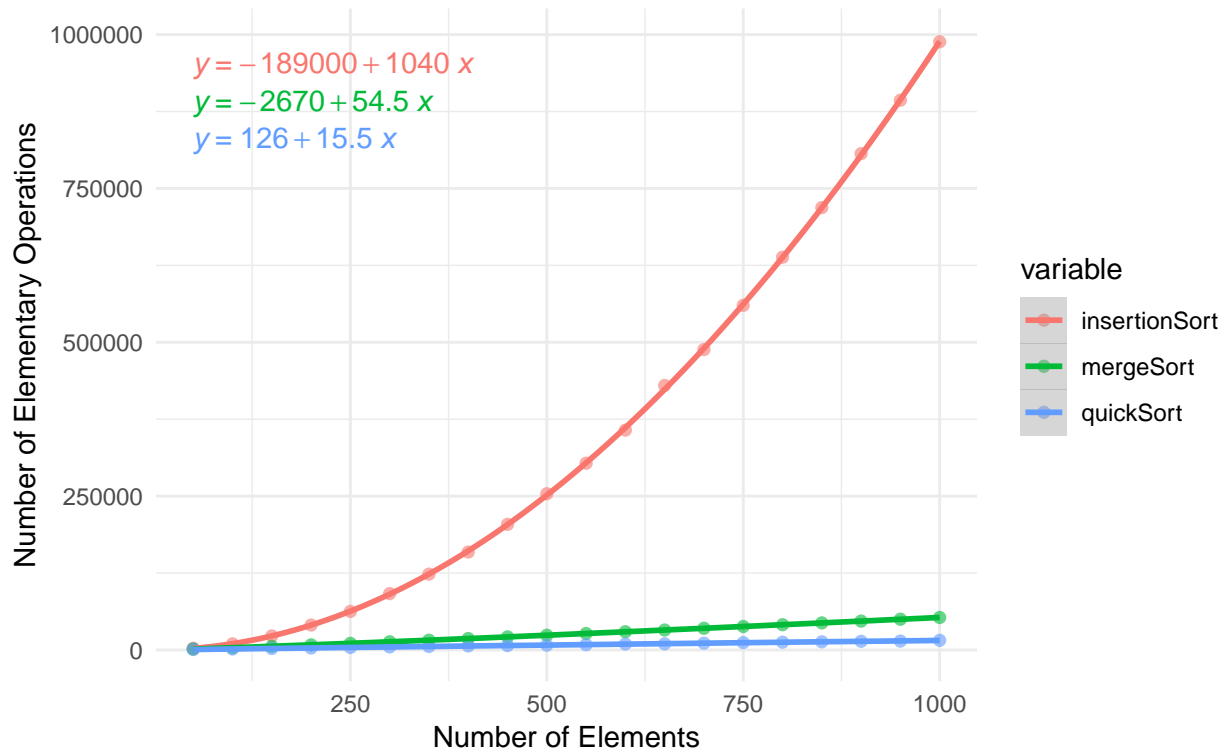
```

df_small <- melt(df_small, id.vars = "ele")
comb_plotter(df_small, "Combined Scatter Plot for small N")

```

Combined Scatter Plot for small N

Time vs Size



Large N

```
df_big <- data.frame(insertionSort = isdf_big[[2]],
                     mergeSort = msdf_big[[2]],
                     quickSort = qsdf_big[[2]],
                     ele = msdf_big[[1]])
df_big
```

##	insertionSort	mergeSort	quickSort	ele
## 1	63201.8	10690.6	4087.4	250
## 2	247054.2	23881.0	7812.5	500
## 3	562903.4	38105.8	12212.7	750
## 4	982349.4	52755.8	15114.9	1000
## 5	1536933.4	68133.0	19045.2	1250
## 6	2234135.0	83752.6	21233.4	1500
## 7	3021561.0	99535.8	27884.4	1750
## 8	3942362.2	115491.0	30466.8	2000
## 9	5041619.4	132025.4	34444.4	2250
## 10	6191853.0	148711.0	39244.8	2500
## 11	7459367.4	165522.2	41003.6	2750
## 12	8883405.8	182385.8	45298.4	3000
## 13	10467540.2	199491.0	50898.0	3250
## 14	12226580.2	216529.4	52182.0	3500
## 15	13932106.2	233678.2	54750.8	3750
## 16	15824639.4	250769.4	60689.0	4000

```
## 17 17852322.2 268493.4 64967.8 4250
## 18 20055156.2 286473.0 66432.4 4500
## 19 22398260.6 304393.8 72392.4 4750
## 20 24713199.8 322325.8 76663.8 5000
## 21 27347217.8 340319.8 77500.8 5250
## 22 30070327.0 358393.0 82246.4 5500
## 23 32735911.0 376500.2 85467.8 5750
## 24 35687765.8 394555.8 93725.0 6000
## 25 38588668.2 413005.0 92000.6 6250
## 26 41808637.0 431148.6 94690.4 6500
## 27 44961291.4 449582.2 97414.2 6750
## 28 48656159.0 467959.4 104127.6 7000
## 29 52160421.0 486335.4 110537.4 7250
## 30 55393425.8 504755.0 112245.2 7500
## 31 59567841.4 523261.8 116673.0 7750
## 32 63386923.4 541760.6 117661.6 8000
## 33 67302632.2 560334.2 120484.8 8250
## 34 71475986.6 579356.2 128940.6 8500
## 35 75883625.4 598517.4 125752.2 8750
## 36 79980967.4 617575.4 135759.2 9000
## 37 84931350.2 636857.0 150196.8 9250
## 38 89409865.4 655948.2 143204.2 9500
## 39 93656091.0 675272.6 150253.4 9750
## 40 98880389.0 694520.2 152601.2 10000
```

```
df_big <- melt(df_big, id.vars = "ele")
comb_plotter(df_big, "Combined Scatter Plot for large N")
```

Combined Scatter Plot for large N

Time vs Size

