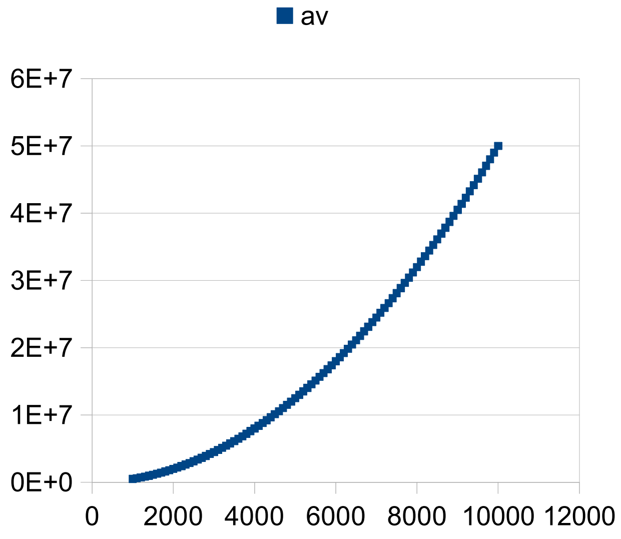
Frederico Castro Banner N03139680
Homework 2 – Estimate Average time of Xsort

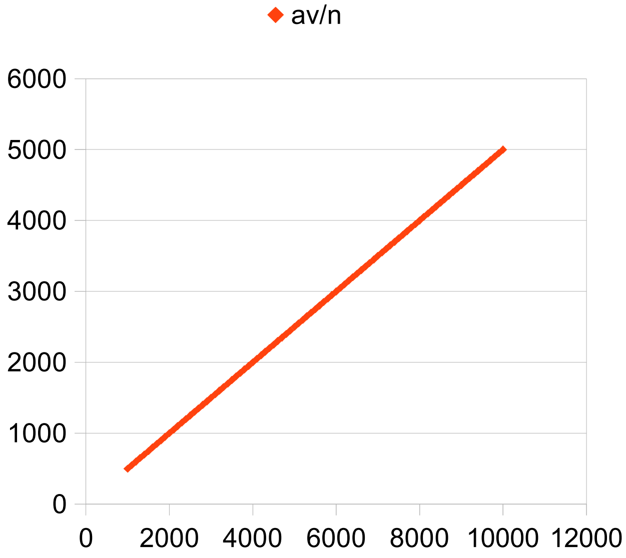
```
class Array
  def swap!(i,j)
    self[i], self[j] = self[j], self[i]
    self
  end
end

def generateRandomPermutation(n)
  sequence = []

  n.times do |i|
    sequence[i] = i
  end

  (n - 1).downto(1) do | index |
    randomIndex = rand(1..index)
    sequence.swap!(index, randomIndex)
  end

  sequence
end

def bSort(list)
  size = list.length
  @comparisonsCounter = 0

  unless size == 1
    pass = 1
    swap = true

    while (pass <= size - 1 && swap) do
      swap = false
      0.upto(size - 1 - pass) do |i|
        @comparisonsCounter += 1
        if list[i] > list[i+1]
          list.swap!(i, i+1)
          swap = true
        end
      end
      pass += 1
    end
  end
  list
end

def insertionSort(list)
  size = list.length
  @comparisonsCounter = 0
  1.upto(size - 1) do |i|
    temp = list[i]
    j = i - 1
    while j >= 0 && list[j] > temp do
      @comparisonsCounter += 1
      list[j+1] = list[j]
      j -= 1
    end

    list[j + 1] = temp
  end
  list
end

def estimateAverageRuntime(xSort)
  averages = []
  (1000..10000).step(100) do |n|
    total = 0

    30.times do |i|
      randomList = generateRandomPermutation(n)
      send(xSort, randomList)
      total += @comparisonsCounter
    end
    averageComparisons = total.to_f/30
    averages << [ n, averageComparisons ]
    printf "."
  end
  averages
end

def prepareResultsToBeExported(results)
  preparedResults = []
  results.each do |result|
    n = result[0]
    average = result[1].to_f
    preparedResults <<  [ n, average, average/n, average/(n*n),
average/(n*Math.log2(n))]
  end
  preparedResults
end

def exportCsv(preparedResults, fileName)
  require 'csv'
  CSV.open("#{fileName}.csv", "w") do |csv|
    csv << ["n", "av", "av/n", "av/n^2", "av/nlogn"]
    preparedResults.each do | result |
      csv << result
    end
  end
end

def mainCall
  begin
    ["bSort", "insertionSort"].each do |xSort|
      xSortAverages = estimateAverageRuntime(xSort)
      xSortResults =
prepareResultsToBeExported(xSortAverages)
      exportCsv(xSortResults, xSort.to_s)
    end
    p "Done"
  rescue
    p "Error"
  end
end

mainCall()
```

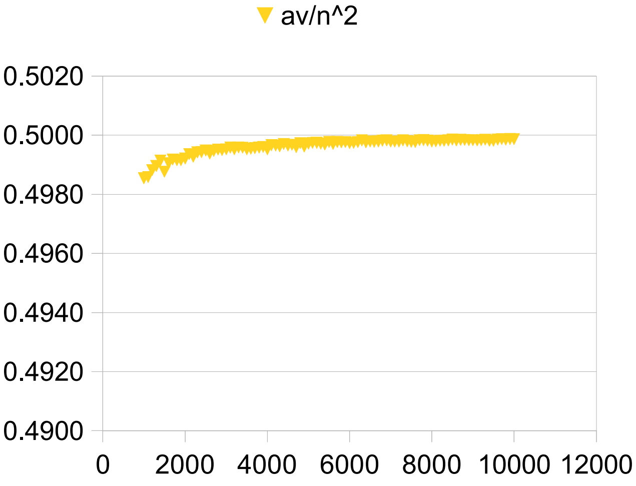
Bubble Sort

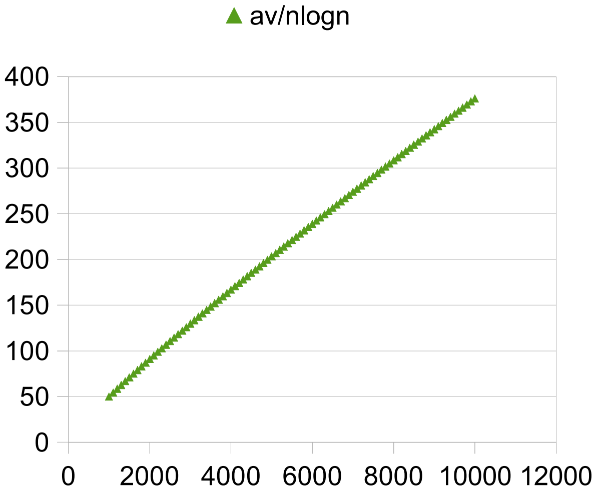
n	av	av/n	av/n^2	av/nlogn
1000	498544	498.54	0.498544	50.03
1100	603283	548.44	0.498581	54.28
1200	718296	598.58	0.498817	58.52
1300	843244	648.65	0.498961	62.71
1400	978307	698.79	0.499136	66.86
1500	1122216	748.14	0.498763	70.91
1600	1277578	798.49	0.499054	75.02
1700	1442625	848.60	0.499178	79.08
1800	1617265	898.48	0.499156	83.09
1900	1801947	948.39	0.499154	87.07
2000	1996841	998.42	0.499210	91.05
2100	2202186	1048.66	0.499362	95.02
2200	2416495	1098.41	0.499276	98.93
2300	2641935	1148.67	0.499421	102.86
2400	2876696	1198.62	0.499426	106.75
2500	3121767	1248.71	0.499483	110.63
2600	3375790	1298.38	0.499377	114.45
2700	3641155	1348.58	0.499473	118.31
2800	3916209	1398.65	0.499516	122.14
2900	4200746	1448.53	0.499494	125.94
3000	4495702	1498.57	0.499522	129.74
3100	4800981	1548.70	0.499582	133.53
3200	5115114	1598.47	0.499523	137.28
3300	5440497	1648.64	0.499587	141.05
3400	5775120	1698.56	0.499578	144.79
3500	6119219	1748.35	0.499528	148.50
3600	6474102	1798.36	0.499545	152.23
3700	6838990	1848.38	0.499561	155.94
3800	7213945	1898.41	0.499581	159.64
3900	7598997	1948.46	0.499605	163.33
4000	7992401	1998.10	0.499525	166.98
4100	8399319	2048.61	0.499662	170.70
4200	8813878	2098.54	0.499653	174.35
4300	9238053	2148.38	0.499624	177.99
4400	9674280	2198.70	0.499705	181.66
4500	10118010	2248.45	0.499655	185.28
4600	10573127	2298.51	0.499675	188.91
4700	11036180	2348.12	0.499601	192.49
4800	11513736	2398.70	0.499728	196.15
4900	11995714	2448.10	0.499613	199.71
5000	12492912	2498.58	0.499716	203.34
5100	12998238	2548.67	0.499740	206.94
5200	13513047	2598.66	0.499743	210.52
5300	14037092	2648.51	0.499718	214.08
5400	14570880	2698.31	0.499687	217.63
5500	15118028	2748.73	0.499770	221.22
5600	15670666	2798.33	0.499702	224.74
5700	16237559	2848.69	0.499771	228.32
5800	16811909	2898.60	0.499759	231.85
5900	17396864	2948.62	0.499766	235.39
6000	17990213	2998.37	0.499728	238.90
6100	18595306	3048.41	0.499739	242.43
6200	19210992	3098.55	0.499766	245.95
6300	19838033	3148.89	0.499824	249.49
6400	20470307	3198.49	0.499763	252.97
6500	21116157	3248.64	0.499791	256.48
6600	21770424	3298.55	0.499780	259.97
6700	22435494	3348.58	0.499788	263.46
6800	23111588	3398.76	0.499818	266.96
6900	23796580	3448.78	0.499823	270.44
7000	24489390	3498.48	0.499783	273.89
7100	25194094	3548.46	0.499784	277.36
7200	25909065	3598.48	0.499789	280.83
7300	26635595	3648.71	0.499824	284.31
7400	27368941	3698.51	0.499798	287.75
7500	28112779	3748.37	0.499783	291.19
7600	28866790	3798.26	0.499771	294.63
7700	29634949	3848.69	0.499830	298.10
7800	30409790	3898.69	0.499832	301.54
7900	31193607	3948.56	0.499817	304.96
8000	31986547	3998.32	0.499790	308.37
8100	32791576	4048.34	0.499795	311.80
8200	33607375	4098.46	0.499812	315.23
8300	34432061	4148.44	0.499812	318.65
8400	35267482	4198.51	0.499823	322.07
8500	36114472	4248.76	0.499854	325.49
8600	36966827	4298.47	0.499822	328.88
8700	37833193	4348.64	0.499844	332.29
8800	38707500	4398.58	0.499839	335.69
8900	39591439	4448.48	0.499829	339.07
9000	40486257	4498.47	0.499830	342.46
9100	41390221	4548.38	0.499822	345.84
9200	42305673	4598.44	0.499831	349.23
9300	43231649	4648.56	0.499846	352.62
9400	44163248	4698.22	0.499810	355.97
9500	45108185	4748.23	0.499814	359.34
9600	46066264	4798.57	0.499851	362.74
9700	47032203	4848.68	0.499864	366.11
9800	48005478	4898.52	0.499849	369.46
9900	48992232	4948.71	0.499870	372.83
10000	49986491	4998.65	0.499865	376.19

Bubble Sort


■ av
6E+7
5E+7
4E+7
3E+7
2E+7
1E+7
0E+0
0
2000
4000
6000
8000
10000
12000


◆ av/n
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000

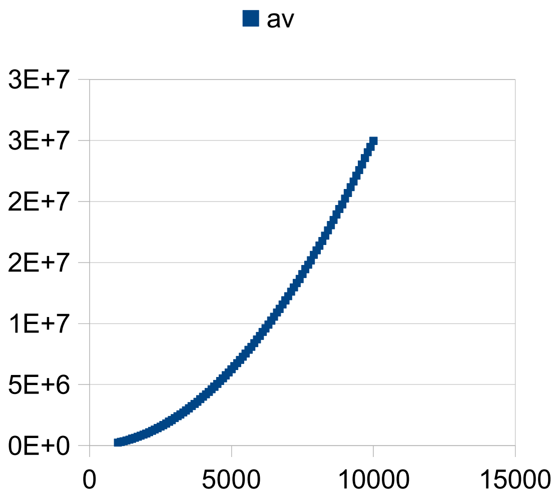

▼ av/n^2
0.5020
0.5000
0.4980
0.4960
0.4940
0.4920
0.4900
0
2000
4000
6000
8000
10000
12000

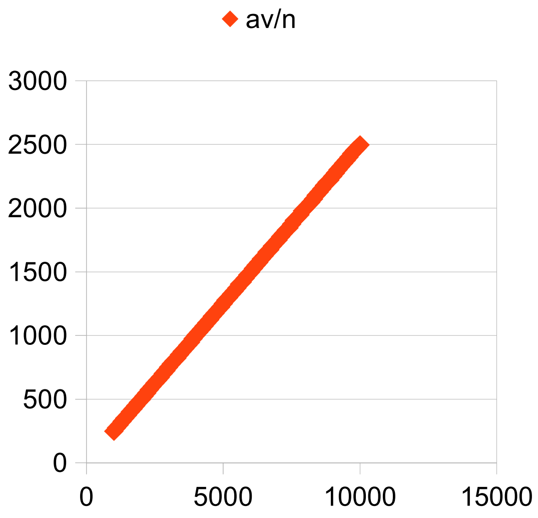

▲ av/nlogn
400
350
300
250
200
150
100
50
0
0
2000
4000
6000
8000
10000
12000

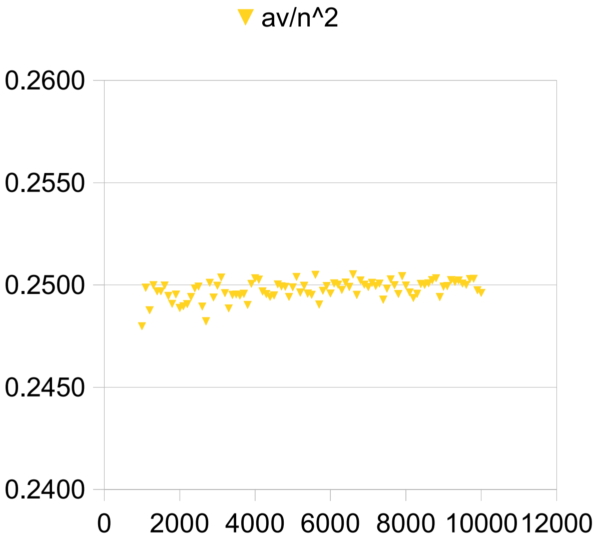
Insertion Sort

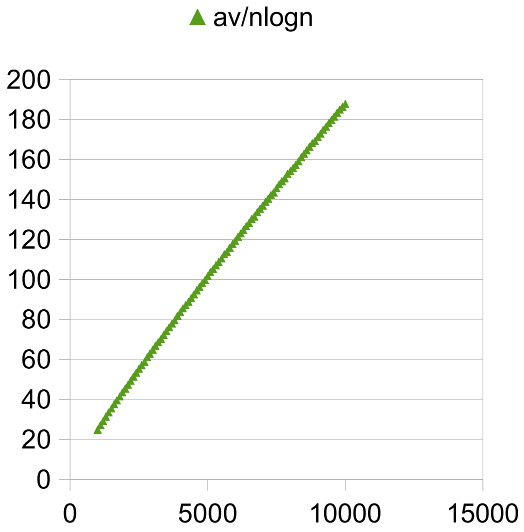
n	av	av/n	av/n^2	av/nlogn
1000	247978	247.98	0.247978	24.88
1100	302335	274.85	0.249864	27.20
1200	358216	298.51	0.248761	29.18
1300	422484	324.99	0.249991	31.42
1400	489393	349.57	0.249690	33.45
1500	561802	374.53	0.249690	35.50
1600	639927	399.95	0.249971	37.58
1700	720939	424.08	0.249460	39.52
1800	807012	448.34	0.249078	41.46
1900	900803	474.11	0.249530	43.53
2000	995494	497.75	0.248873	45.39
2100	1097896	522.81	0.248956	47.37
2200	1205464	547.94	0.249063	49.35
2300	1319395	573.65	0.249413	51.37
2400	1439033	599.60	0.249832	53.40
2500	1562000	624.80	0.249920	55.35
2600	1682836	647.24	0.248940	57.05
2700	1809531	670.20	0.248221	58.80
2800	1960808	700.29	0.250103	61.15
2900	2097362	723.23	0.249389	62.88
3000	2249683	749.89	0.249965	64.92
3100	2406035	776.14	0.250368	66.92
3200	2555899	798.72	0.249599	68.60
3300	2710062	821.23	0.248858	70.26
3400	2884341	848.34	0.249510	72.31
3500	3056796	873.37	0.249534	74.18
3600	3233247	898.12	0.249479	76.02
3700	3416536	923.39	0.249564	77.90
3800	3595884	946.29	0.249022	79.57
3900	3803385	975.23	0.250058	81.75
4000	4005276	1001.32	0.250330	83.68
4100	4206802	1026.05	0.250256	85.49
4200	4404370	1048.66	0.249681	87.13
4300	4614168	1073.06	0.249549	88.90
4400	4828558	1097.40	0.249409	90.67
4500	5052050	1122.68	0.249484	92.51
4600	5290515	1150.11	0.250024	94.52
4700	5520945	1174.67	0.249930	96.30
4800	5757960	1199.57	0.249911	98.09
4900	5988511	1222.15	0.249417	99.70
5000	6247197	1249.44	0.249888	101.68
5100	6512715	1277.00	0.250393	103.68
5200	6749955	1298.07	0.249629	105.16
5300	7021428	1324.80	0.249962	107.08
5400	7277712	1347.72	0.249579	108.70
5500	7548082	1372.38	0.249523	110.45
5600	7855690	1402.80	0.250500	112.66
5700	8091383	1419.54	0.249042	113.77
5800	8400442	1448.35	0.249716	115.85
5900	8700736	1474.70	0.249949	117.73
6000	8984953	1497.49	0.249582	119.31
6100	9305215	1525.45	0.250073	121.31
6200	9610622	1550.10	0.250016	123.04
6300	9912657	1573.44	0.249752	124.67
6400	10244715	1600.74	0.250115	126.60
6500	10558392	1624.37	0.249903	128.24
6600	10912459	1653.40	0.250516	130.31
6700	11200742	1671.75	0.249515	131.53
6800	11570114	1701.49	0.250219	133.65
6900	11903376	1725.13	0.250018	135.28
7000	12245078	1749.30	0.249900	136.95
7100	12607432	1775.69	0.250098	138.80
7200	12957477	1799.65	0.249951	140.45
7300	13326138	1825.50	0.250068	142.24
7400	13650435	1844.65	0.249277	143.52
7500	14052437	1873.66	0.249821	145.55
7600	14456175	1902.13	0.250280	147.55
7700	14822043	1924.94	0.249992	149.10
7800	15182896	1946.53	0.249554	150.55
7900	15628951	1978.35	0.250424	152.80
8000	15998412	1999.80	0.249975	154.24
8100	16378917	2022.09	0.249641	155.74
8200	16767024	2044.76	0.249361	157.27
8300	17192599	2071.40	0.249566	159.11
8400	17643310	2100.39	0.250047	161.12
8500	18062587	2125.01	0.250001	162.80
8600	18496035	2150.70	0.250082	164.55
8700	18940032	2177.02	0.250232	166.35
8800	19385483	2202.90	0.250329	168.12
8900	19755565	2219.73	0.249407	169.19
9000	20243150	2249.24	0.249915	171.23
9100	20697715	2274.47	0.249942	172.94
9200	21180320	2302.21	0.250240	174.84
9300	21634884	2326.33	0.250143	176.46
9400	22109891	2352.12	0.250225	178.21
9500	22568722	2375.65	0.250069	179.79
9600	23040417	2400.04	0.250005	181.43
9700	23549473	2427.78	0.250287	183.31
9800	24038575	2452.92	0.250298	185.01
9900	24477409	2472.47	0.249744	186.27
10000	24961238	2496.12	0.249612	187.85

Insertion Sort


■ av
3E+7
3E+7
2E+7
2E+7
1E+7
5E+6
0E+0


◆ av/n


▼ av/n^2


▲ av/nlogn