Student ID#: 580012-F  
Alexander Zen Taniguchi-Wiegman  
Week 10 Exercises

1st Exercise) Given the array, a= [1,4,2,9,8,3,2,6,4]. Determine behavior at each step for…  
 a) the simple sort

**1**, 4, 2, 9, 8, 3, 2, 6, 4

1, **2**, 4, 9, 8, 3, 2, 6, 4

1, 2, **2**, 4, 9, 8, 3, 6, 4

1, 2, 2, **3**, 4, 9, 8, 6, 4

1, 2, 2, 3, **4**, 9, 8, 6, 4

1, 2, 2, 3, 4, **4**, 9, 8, 6

1, 2, 2, 3, 4, 4, **6**, 9, 8

1, 2, 2, 3, 4, 4, 6, **8**, 9

1, 2, 2, 3, 4, 4, 6, 8, **9**  
  
 b) the merge sort

1, 4, 2, 9, 8, 3, 2, 6, 4

14, 29, 38, 26, 4 (says copy, not merge, so I copy it over and leave it alone)

1249, 2368, 4

122344689

2. Complete the simple sort algorithm

# This code is written by Alex T.W., with quite some assistance from Stack Overflow user "unj2".

#It is meant to answer exercise 2 question of Week 10, Prof. Kakimura's Information class.

# In the interest of not plagiarizing, I would like to say that in my confusion I looked up this topic on Stack Overflow and came across this page (https://stackoverflow.com/questions/855693/is-there-a-better-way-to-find-the-location-of-a-minimum-element-in-an-array) with rather inspiring code.

def min\_index(a,i)

minimum = 0

for i in 1..(a.length())

if array[i]<array[0]

minimum = i

end

end

return minimum

end

def simplesort(a)

for i in 0..(a.length()-1)

k = min\_index(a,i)

v = a[i]

a[i] = a[k]

a[k] = v

end

a

end

#This should allow for simple-sort method sorting of a positive number array (I might need to find a more foolproof method with negative numbers.)

1. Complete the merge sort algorithm

load("./is\_even.rb")

load("./make2d.rb")

#Creative re-use of make2d's array nature to call the new array with make2d instead of the more regular term, array.new

def merge(a,b) #Important to know a and b are already sorted.

c = make1d(a.length()+b.length())

ia=0

ib=0

ic=0

while ia < a.length() && ib < b.length()

if a[ia] < b[ib]

c[ic] = a[ia]

ia = ia + 1

ic = ic + 1

else

c[ic] = b[ib]

ib = ib + 1

ic = ic + 1

end

end

c+a+b #Take what you know have of c + a or b, the blank one + the other of a or b, the one with the remainder of your sorted terms.

c

end

def mergesort(a)

n = a.length()

from = make2d(n,1)

for i in 0..(n-1)

from[i][0] = a[i]

end

while n > 1

to = make1d((n+1)/2)

for i in 0..(n/2-1)

to[i] = merge(from[i\*2], from[i\*2+1])

end

if !is\_even(n)

to[(n+1)/2-1]=from[n-1]

end

from=to

n=(n+1)/2

end

from[0]

end

1. Compare the actual computational time

In the interest of being truthful, and because I would not have been able to do this problem without a lot of help. I will leave this problem as be. I had a cold for a long time, and by the time I was able to return to this final, the answer had just been published in Week 12 (Lec12Answers.pdf, pages 12-17).