COMS W1004              MW 4:10-5:25 PM

Columbia University                          Spring 2014

**Homework 1: Problem Set**

**February 3, 2014**

Your CUNIX ID: ami2119

Your Last Name: Iwamizu

Your First Name: Akiko

**Circle the range that includes your UNI:**

Group 1 (aa3473-amb2368)  Group 8 (kh2633-mc3834)

**Group 2 (amg2300-bac2176)** Group 9 (mcs2225-nam2182)

Group 3 (bbcb2137-cyb2105) Group 10 (nay2109-rdy2104)

Group 4 (db2941-erw2138) Group 11 (rhj2002-sec2161)

Group 5 (es2680-ip2273) Group 12 (sfm2123-tla2119)

Group 6 (it2217-jmf2211) Group 13 (tlb2145-zn2116)

Group 7 (jmg2227-kgl2111)

**Problem Set 1**: Chapter 1: 10, 11; Chapter 2: 9, 14, 20, 25; Chapter 3: 9, 14, 28, 36

**Chapter 1: 10, 11**

10) a. Go through this algorithm using the input values 20 and 32. After each step of the algorithm is completed, give the values of I, J, and R. Determine the final output of the algorithm.

Steps:

1. I = 32 , J = 20 , R = ?
2. I = 32 , J = 20 , R = 12
3. I = 20 , J = 12 , R = 12
4. I = 20 , J = 12 , R = 8
5. I = 12 , J = 8 , R = 8
6. I = 12 , J = 8 , R = 4
7. I = 8 , J = 4 , R = 4
8. I = 4 , J = 4 , R = 4
9. I = 4 , J = 4 , R = 0
10. **J = 4**

b. Does the algorithm work correctly when the two inputs are 0 and 32? Describe exactly what happens, and modify the algorithm so that it gives an appropriate error message.

Steps:

1. I = 32 , J = 0 , R = ?
2. I = 32 , J = 0, R = (32/0) = ERROR

The algorithm does not work correctly when J = 0, because it gives an error when dividing by 0. Thus, a possible modification would be adding a step 3 saying:

1. If J = 0, then print “Error: Division by 0” and STOP.

11) If a computer could analyze 10,000,000 separate paths per second, how long would it take to determine the optimal route for visiting these 25 cities? On the basis of your answer, do you think this is a feasible algorithm? If it is not, can you think of a way to obtain a reasonable solution to this problem?

I know there are 25 cities that must be visited and thus 25! possible paths that can be chosen. I also know that the computer can analyze 10,000,000 separate paths per second (pps), which can be simplified to 107. Thus, let us calculate how many years that would take to complete:

* 25! = 1.55 x 1025 possible paths

(1.55 x 1025 paths/107 pps) \* (1pps/ 60 secs) (1 sec/60 mins) \* (1 min/24 hrs) \* (1 hr/365 days)

**= 4.9186 x 1010 years**

Hence, on this basis of my answer, this is certainly not a feasible algorithm. A reasonable solution to this problem would be to look for the shortest path initially and take that shortest path to a city you haven’t already visited yet, and repeat for the other 24 cities.

**Chapter 2: 9, 14, 20, 25**

4) (Answered this problem because it is the basis of problem 9) Write an algorithm that gets the price for the item A plus the quantity purchased. The algorithm prints the total cost, including a 6% sales tax.

Steps:

1. Let A = price of the item and x = quantity of that item purchased
2. Let y be the subtotal such that y = (A \* x)
3. Take 6% sales tax of y such that total cost = (0.06 \* y)
4. Print “total cost”
5. STOP

9) Modify the sales computation algorithm of Exercise 4 so that after finishing the computation for one item, it starts on the computation for the next. This iterative process is repeated until the total cost exceeds $1,000.

1. Let global cost = 0
2. Let A = price of the item and x = quantity of that item purchased
3. Let y be the subtotal such that y = (A \* x)
4. Take 6% sales tax of y such that Atotal cost = (0.06 \* y) and add this amount to global cost
5. If global cost is ≤ $1,000, then repeat steps 2-4 for another item and quantity
6. If global cost is > $1,000, then STOP

14) Use the Find Largest algorithm to help you develop an algorithm to find the median value in a list containing N unique numbers. The median of N numbers is defined as the value in the list in which approx. half the values are smaller than it. If N is an even value, then the number of values larger than the median will be one greater than the number of values smaller than the median.

Steps:

Get a value for n, the size of the list

Get values for A1, A2… An, the list to be searched

Set the value of *largest so far* to A1

Set the value of *location* to 1

Set the value of i to 2

While (i ≤ n) do

If Ai > *largest so far* then

Set *largest so far* to Ai

End of the loop

Remove *largest so far* and repeat the steps times if n is odd and times if n is even

Print out the values of the or (depending on n) *largest so far* valueand *location*

STOP

20) Design an algorithm that is given a positive integer N and determines whether N is a prime number, that is, not evenly divisible by any value other than 1 and itself. The output of your algorithm is either the message ‘not prime,’ along with a factor of N, or the message ‘prime.’

Steps:

1. If N is 1 or 2, then print “prime”
2. If N is even and not 2, then print “not prime with a factor of 2”
3. If N is odd, then assume N is prime
4. Suppose n divides N and let n ∈ℕ+
5. Assume 2 < n < and odd
6. If n divides N holds, then this gives a contradiction to step 3 and thus, print “not prime with a factor of n”
7. If n divides N does not hold, then print “prime”
8. STOP

25) Write an algorithm to read in a sequence of values V ≥ 0, one at a time, and determine if the list contains at least one adjacent pair of values that are identical. The end of the entire list is marked by the special value V = -1. The output of your algorithm should be a ‘Yes’ if there is at least one pair of adjacent numbers that are equal (the 7s). The output of your algorithm should be a ‘No’ if there are no adjacent pairs that are equal. You may assume in your solution that there are at least two numbers in the list.

Steps:

1. Let i = 1
2. If Ai = Ai+1 , then print “Yes”
3. If Ai ≠ Ai+1 , then repeat and increment i++ until Ai+1 = -1
4. If Ai ≠ Ai+1 still holds then print “No”
5. STOP

**Chapter 3: 9, 14, 28, 36**

9)

a) 4, 8, 2, 6

1. 4, 2, 6, 8

2. 2, 4, 6, 8

=> 2 exchanges

b) 12, 3, 6, 8, 2, 5, 7

1. 3, 6, 8, 2, 5, 7, 12

2. 3, 6, 2, 5, 7, 8, 12

3. 3, 2, 5, 6, 7, 8, 12

4. 2, 3, 5, 6, 7, 8, 12

=> 4 exchanges

c) D, B, G, F, A, C, E, H

1. B, D, G, F, A, C, E, H

2. B, D, F, A, C, E, G, H

3. B, D, A, C, E, F, G, H

4. B, A, C, D, E, F, G, H

5. A, B, C, D, E, F, G, H

=> 5 exchanges

Comparing to practice problem 3.3.3:

a) 4, 8, 2, 6 |

1. 4, 6, 2 | 8

2. 4, 2 | 6, 8

3. 2 | 4, 6, 8

4. | 2, 4, 6, 8

=> 3 exchanges

b) 12, 3, 6, 8, 2, 5, 7 |

1. 7, 3, 6, 8, 2, 5 | 12

2. 7, 3, 6, 5, 2 | 8, 12

3. 2, 3, 6, 5 | 7, 8, 12

4. 2, 3, 5 | 6, 7, 8, 12

5. | 2, 3, 5, 6, 7, 8, 12

=> 4 exchanges

c) D, B, G, F, A, C, E, H

1. D, B, G, F, A, C, E | H

2. D, B, E, F, A, C | G, H

3. D, B, E, C, A | F, G, H

4. D, B, A, C | E, F, G, H

5. C, B, A | D, E, F, G, H

6. A, B, | C, D, E, F, G, H

7. | A, B, C, D, E, F, G, H

=> 5 exchanges

14) Algorithms A and B perform the same task. On input of size n, algorithm A executes 0.003n2 instructions, and algorithm B executes 243n instructions. Find the appropriate value of n above which algorithm B is more efficient. (You may use a calculator or spreadsheet)

243n = 0.003n2

243 = 0.003n

= n

81,000 = n (This tells us that the formulas for A and B intersect when n=81,000)

Thus, B is more efficient for **n > 81,000**.

28) If the list size is n = 100,000, about how many worst-case searches must be done before the second alternative is better in terms of number of comparisons?

Using the formulas for the worst-case searches of sequential search vs. sort list then binary search I solve for p in the following equation:

[p (n) – p (lgn)] = n2

p ( n – lgn ) = n2

p = = = **100, 017**

36)

a) Let n have the value 4. Write the values printed out by this algorithm.

j = 2, j = 1, j = 2, j = 1, j = 2, j = 1, j = 2, j = 1

=> **8** values printed

b) Let n have the value 8. Write the values printed out by this algorithm.

j = 4, j = 2, j = 1, j = 4, j = 2, j = 1, j = 4, j = 2, j = 1, j = 4, j = 2, j = 1, j = 4, j = 2, j = 1,

j = 4, j = 2, j = 1, j = 4, j = 2, j = 1, j = 4, j = 2, j = 1

=> **24** values printed

c) Which of the following best describes the efficiency of this algorithm, where the “work unit” is printing a value?

**Θ (n lg n)** because Θ (4 lg 4) = 8 and Θ (8 lg 8) = 24, which is the exact number of values that were printed by the algorithm.

d) How many work units would you expect this algorithm to do if n = 16?

Since we determined that the work unit is Θ (n lg n), if n = 16 then the result would be

**Θ (16 lg 16) = 64.**