10/24/2019

**Name: Getaneh Tafere**

**ID: 110226**

**CS 311 Discrete Math and Data Structures:**

**Homework4**

4. Describe an algorithm that takes as input a list of n integers and produces as output the largest difference obtained by subtracting an integer in the list from the one following it.

**function largestDifference(a1, a2, a3, …, an) {**

**let maxDiff = a2 - a1**

**let diff;**

**for(let i=2; i< a.length; i++){  
diff=ai - ai-1**

**}**

**If(maxDiff<diff){  
maxDiff=diff;**

**}**

**Return** **maxDiff;**

13. List all the steps used to search for 9 in the sequence 1, 3, 4, 5, 6, 8, 9, 11 using

a) a linear search.

To search 9 we compare with each elements starting from the first. Like is 1===9? no

So, continue to the next element, is 3===9? no … finally is 9===9? Yes.

b) a binary search.

By dividing sequences in two halves 1, 3, 4, 5 and 6, 8, 9, 11. compare the largest one from first half, that means from the largest half the largest number is 5, since 5<9, it takes the second half and divided in to two half 6, 8 and 9, 11 and compare the largest one from first half, since 8<9 it takes the second half and divided in to two halves 9 and 11. Now compare 9 = 9, we found 9.

35. Use the **bubble sort** to sort 3, 1, 5, 7, 4, showing the lists obtained at each step.

It compares two adjacent elements and sort them

First Pass 3 > 1, they need to be interchanged. 1, 3, 5, 7, 4

Then 2nd and 3rd elements. 3 < 7 not need to be interchanged. 1, 3, 5, 7, 4

Next 3nd and 4rth5<7 they no need to be interchanged. 1, 3, 5, 7, 4

Next 4nd and 5rth7>4 they need to be interchanged. 1, 3, 5, 4, 7.

Second Pass the same thing compares starting from first to end.

Third Pass the same thing compare starting from first to end and making sure…

39. Use the **insertion sort** to sort the list in Exercise35, showing the lists obtained at each step.

It compares the second element to the first element and correctly sort these two numbers.

First Pass: the first two elements 3 > 1, 1 should be before the three in the data set. 1, 3, 5, 7, 4.

Second Pass: we want to insert 5 into already sorted parts, since 5 > 1 and 5> 3 it should be after 3….

And continue as the same manner. Finally sorts like 1, 3, 4, 5, 7

52. Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for

a) 87 cents.

**Quarter: 25 cent**

**Dime: 10 cent**

**Nickle: 5 cent**

**Penny: 1 cent**

**Total number n= 87 , since n> 25 we add 1 quarter and declare the total amount**

**n= nold- 25 = 87 – 25 =62**

**And again n>25 so add 25 and n= nold- 25 = 62 – 25 =37**

**And again n>25 so, add 25 and n= nold- 25 = 37 – 25 =12 now n < 25**

**n> 10, thus we add 1 dime and decrease the total amount by 10 n= nold- 10 = 12-10 =2**

**now n< 10 and also n< 5 but n>=1, thus we add 1 penny and decrease the total amount by 1, n= nold- 1= 2-1= 1. n>=1, thus we add 1 penny and decrease the total amount by 1, n= nold- 1= 1-1= 0 Conclusion 87 cents is 3 quarters, 1 dime, 0 nick and 2 pennies**

57. Use Algorithm 7 to schedule the largest number of talks in a lecture hall from a proposed set of talks, if the starting and ending times of the talks are

**1. 9:00a.m.and9:45a.m.;**

**2. 9:30 a.m. and 10:00 a.m.;**

**3. 9:50 a.m. and 10:15 a.m.;**

**4. 10:00 a.m. and 10:30 a.m.;**

**5. 10:10 a.m. and 10:25 a.m.;**

**6. 10:30 a.m. and 10:55 a.m.;**

**7. 10:15 a.m. and 10:45 a.m.;**

**8. 10:30 a.m. and 11:00 a.m.;**

**9. 10:45 a.m. and 11:30 a.m.;**

**10. 10:55 a.m. and 11:25 a.m.;**

**11. 11:00 a.m. and 11:15a.m**

**So starting from the first lecture we choose the next starting time later than the finishing time. 1, 3, 6 and 11 are schedule the largest number of talks**

1. Determine whether each of these functions is O(x).

a) f(x)=10: O(x)

b) f(x)=3x+7 : O(x)

c) f(x)=x2+x+1:O(x)

d) f(x)=5logx : O(x)

e) f(x)=⌊x⌋ f) f(x)=⌈x/2⌉ : O(x)

3. Use the deﬁnition of “f(x) is O(g(x))” to show that x4+9x3+4x+7 is O(x4).

**F(x)= x4+9x3+4x+7 we will choose k=9, and thus x> 9. |f(x)|= | x4+9x3+4x+7|**

**<=| x4|+|9x3|+|4x|+|7|= x4+9x3+4x+7 < x4+ x4+ x4+ x4 =4x4 =4|x4|**

**So the bog-O f(x) = O(x4)**

28. For each function in Exercise 1, determine whether that function is Theta(x) and whether it is Omega(x).

a) f(x)=10: not

b) f(x)=3x+7: Yes …

1. Give a big-O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm. t :=0 for i :=1 to 3 for j :=1 to 4 t :=t + ij: **O(1)**

7. Suppose that an element is known to be among the ﬁrst four elements in a list of 32 elements. Would a linear search or a binary search locate this element more rapidly?

Linear search

37. Find the complexity of a brute-force algorithm for scheduling the talks by examining all possible subsets of the talks. [Hint: Use the fact that a set with n elements has 2n

subsets.]

**Total possible subset is 2n.  time complexity is O(n2) for each set. So, there are O(2n\*n2).**