Rosa Cho  
7/9/2021

Complete hw by giving answers to the questions below, then upload the document in Canvas

Exercise 2-3. Give a) an example of a valid instance of the summation problem that is not a valid instance of the minimum problem, and b) vice-versa.

Answer:

a) An example of a summation problem that ***is not*** a valid instance of a minimum problem would be adding up all the elements in X.

b) An example of a summation problem that ***is*** a valid instance of a minimum problem would be finding the least value number in the elements in X.

Exercise 2-4. Each of the following snippets of pseudocode fails to live up to all of the clarity, correctness, and termination requirements of algorithms. In each case, describe the problem, and then rewrite the pseudocode as a proper algorithm. Hint: consult the checklist in section 2.3.

|  |  |
| --- | --- |
| a)  def contains\_zero(S):  for x in S:  if x == 0:  return True  else:  return False | b)  def keep\_positives(S):  if len(S) == 0:  return 0  else:  result = []  for x in S:  if x > 0:  result.add(x)  return result |

Answer:

|  |  |
| --- | --- |
| a)  def contains\_zero(int S):  for int x in S:  if x == 0:  return true  else:  return false | b)  def keep\_positives(int S):  if len(S) == 0:  return 0  else:  int result = []  for int x in S:  if x > 0:  result.add(x)  return result |

Exercise 2-5. Write a problem definition for each of the following problems.

(d) determining whether two strings are identical  
(e) determining whether a sequence contains entirely positive numbers  
(f) determining whether a sequence is in strictly increasing order (i.e. each number is greater than the last)  
(g) concatenating two sequences into one larger sequence

Answer:

(d) If string\_1 is identical to string\_2, return true; else, if string\_1 is not identical to string\_2 then return false.

(e) If the sequence is full of numbers that are all greater than 0, return true; else if there is at least one element that is less than or equal to 0, return false.

(f) If the sequence is in increasing order in which an element is followed by another element bigger than it, return true; else, if there is at least one element that isn’t followed by another element greater than itself, return false.

(g) If there are two sequences, we combine them into a larger sequence by merging them.

Exercise 3-2. Brute force password guessing. Each of the following is a different kind of password. An attacker can always try to bypass password security by guessing every possible password that might exist. In the worst case (for the attacker), the correct password is the last one that they guess. So increasing the number of potential passwords makes password protection more effective. For each password type, compute the number of passwords of that type that exist. Justify your answers (show your work).

(c) Eight lower-case letters, upper-case letters, or digits.

(d) Nine letters.

Answer:

c)

(\_ + \_ + \_+ \_+\_+\_+\_+\_) + (\_+\_+\_+\_+\_+\_+\_+\_) + (\_+\_+\_+\_+\_+\_+\_+\_) =   
If we allow repeatable elements, then by computing the probability of each combination, we get \_.

d)

\_+\_+\_+\_+\_+\_+\_+\_+\_ =   
Again, if we allow repeatable elements and combinations of both upper-case and lower-case letters, we get \_.

Exercise 3-6. Let g(n) = n. Is it true that g(n) O(n2)? Why or why not?

Answer: No. While if n = 1 or n =0, then n=n^2 would hold true. Otherwise, it is not an inclusive statement.

Exercise 3-7. Evaluate each of the following complexity functions for n = 23 = 8,n = 27 = 128, and n = 210 = 1024. The goal of this exercise is to get a feel for the relative rates of growth of some common efficiency classes.  
(a) log2 n

(b) n  
(c) n2

(d) n3  
(e) 2n

(f) n!

Answer:

a) true  
b) false  
c) false  
d) false  
e) false  
f) false

Exercise 3.13 Rank the following functions by their order of growth (i.e. asymptotic efficiency class), from slowest-growing to fastest-growing.  
(a) n2

(b) log n  
(c) 4log n

(d) n log n  
(e) log log n

(f) n  
(g) 2n

(h) n!

Answer:

f, a, h, g, c, b, e, d