ORDRE DES INGÉNIEURS DU QUÉBEC

MAY 2019 SESSION

Open-book examination
Calculators: only authorized models
Duration: 3 hours

14-IF-A4 Program Design and Data Structures

Programming languages are C, C++ and Java.

1. [10 points]
   Design and implement in pseudocode or in code an algorithm that gets as input a list of n integers $L_1, L_2, ..., L_n$ as well as a special value sum. Your algorithm must locate a pair of values in the list that sum to the value sum. For example, if your list of values is 3, 8, 13, 2, 17, 18, 10, and the value of sum is 20, then your algorithm should output "either" the two values 2 and 18, "or" the two values 3 and 17. If your algorithm cannot find any pair of values that sum to the value sum, then it should print out the message "sorry, there is no such pair of values".
   Analyze the efficiency of your algorithm (consider the comparison as the main operation), and give the time complexity in θ notation.
2. [10 points]
   Consider the four following pieces of pseudocode. In each case,
   (a) Give the number of times "hello" will be printed;
   (b) Generalize giving the number of times as a function of n;
   (c) Express this in θ notation in terms of n.
   (d) Give an explanation if it is not possible to answer b or c.

   (a) code

```
set n = 4
set i = n
while (i>=1)
   set j = 1
   while (j<=n)
      print "hello"
```

```
      j=j+1
   end while
   i = i+1
endwhile
```

(b) code

```
set n = 4
set i = 1
while i<= n
   set j = 1
   while j<=i
     print "Hello"
     j=j+1
   endwhile
   i=i+1
endwhile
```

(c) code

```
set n = 8
set i = 1
while i<= n
  print "Hello"
  i=i*2
endwhile
```

(d) code

```
set n = 4
set i = 1
while i not equal to n
  print "Hello"
  i=i+2
endwhile
```

3. [10 points]
   Simplify following expressions by using De Morgan law:
   — `!(x > 0 && y > 0 )`
   — `!(x != 0 || y != 0 )`
   — `!(x  % 4 != 0 ) || !(x % 100 == 0 && x % 400 == 0)`
   — `!(x > 2 && x <= 5 )`
   — `!(!(x > 2) ||!(x <= 5) )`
4. [10 points]
   Prove by induction that: if $0 \leq a \leq 1$, then $(1-a)^n \geq 1 - na$.
5. [10 points]
   Write a fonction to find all permutations of the letters in a particular string s.

6. [10 points]
   Is $(x+y)^2 = O(x^2+y^2)$ **true** or **false**. Explain how you can arrive to your conclusion.

7. [10 points]
   Suppose you are given a source string $S[0..n-1]$ of length n, consisting of symbols a and b.

   Suppose further that you are given a pattern string $P[0..m-1]$ of length $m \ll n$, consisting of symbols a, b, and $*$, representing a pattern to be found in string S. The symbol $*$ is a "wild card" symbol, which matches a single symbol, either a or b. The other symbols must match exactly.

   The problem is to output a sorted list M of valid "match positions", which are positions j in S such that pattern P matches the substring $S[j..j+|P|-1]$. For example, if $S = ababbab$ and $P = ab*$, then the output M should be $[0, 2]$.

   Describe a straightforward, naïve algorithm to solve the problem. Your algorithm should run in time $O(nm)$.

8. [10 points]
   Show how to multiply the complex numbers $a+bi$ and $c+di$ using only three multiplications of real numbers. The algorithm should take a, b, c, and d as input and produce the real component $ac?bd$ and the imaginary component $ad+bc$ separately.

9. [10 points]
   For each expression, give the values of variables or indicate that there is an error.

   (a) **int** a = (**int**) (2.3 + 5.2 − 20 % 3 / 18 / 12) + 7 ∗ 2 / 50;

   (b) **double** b = (**double**) (2.3d + 5) − 20 + 3.4 / (18.1d / 12.4);

   (c) **float** c = (2.3f + 5) − (**float**) (20.5d / 3.2) ∗ 18 / 12;

   (d) **int** d = (**int**)(2.3f + 5) − 20 / 3 / ((**int**) (18 % 12));

10. [10 points]
    For each expression, give the values of variables or indicate that there is an error.

    (a) **boolean** aa = (25 <= 21) || !((3 − 12) < 3 || **false**) & (**true** && **false**);

    (b) **boolean** bb = !(18 − 3 < 15) && (18 − 2 < 15) && (1 < 39 % 2 − 1);

    (c) **boolean** cc = !(18 − 17 < 15) | !!(18 − 2 < 15) || !(15 > 37 / 2 − 1);

    (d) **boolean** dd = (((**double**) 2.3 + 5) == (**double**) 7.3) || (!**true**);

End of exam.