ORDRE DES INGÉNIEURS DU QUÉBEC

NOVEMBER 2018 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]
   For each of the following code segments, find a function f(n) with the property that the number of statements executed by the segment is O(f(n)). Your function f(n) should be as simple as possible.
   — Segment 1:
   ```
   for(int i = 0, Sum = 0; i < n; i++)
   Sum++;
   ```
   — Segment 2:
   ```
   for (int i = 0, Sum = 0; i < n; i++)
   for (int j = 0; j < n; j++)
   Sum++;
   ```
   — Segment 3 :
   ```
   for (int i = 0, Sum = 0; i < n; i++)
   Sum++;
   for (int j = 0; j < n * n; j++)
   Sum++;
   ```
   — Segment 4 :
   ```
   for (int i = 0, Sum = 0; i < n; i++)
   for (int j = 0; j < i; j++)
   Sum++;
   ```
   — Segment 5 :
   ```
   for (int i = 0, Sum = 0; i < n; i++)
   for (int j = 0; j < i * i; j++)
   for (int k = 0; k < j ; k+)
   Sum++;
   ```

2. [10 points]

   Give a code to implement a chained list, then give the code for the methods or (functions) for eliminateDuplicates of the list and invert the list.

   For the first method, the list (a, b, a, a, b, c, a) returns (a, b, c).

   For the second method, the list (a, b, c, d, e) returns (e, d, c, b, a).

3. [10 points]

   There are eight small islands in a lake, and the state wants to build seven bridges to connect them so that each island can be reached from any other one via one or more bridges. No connection with the lake border is allowed. The cost of constructing a bridge is proportional to its length. The distances between pairs of islands are given in the following table.

	1	2	3	4	5	6	7	8
1	-	248	210	340	280	200	345	120
2	-	-	265	175	215	180	185	155
3	-	-	-	260	115	350	435	195
4	-	-	-	-	160	330	295	230
5	-	-	-	-	-	360	400	170
6	-	-	-	-	-	-	175	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

   Find which bridges to build so that the total construction cost is minimum.

   Give the algorithm name that you used, and indicate how to find the solution.

4. [10 points]

   For each of the following statements, indicate whether it applies to a precondition or a postcondition.

   (a) Is the responsibility of the client.

   (b) The algorithm implementing the method must make sure that it gets satisfied.

   (c) Must be true when the method is invoked.

   (d) Guaranteed true when the method completes.

5. [10 points]

   Suppose that the array A contains n elements where each can be red, white or blue. This array must be sorted to obtains first all red elements, then all white and at the end all blue. There are only two operations for this array :

   (a) read(A, i) this gives the color of the ith element of A (same that A[i]).

   (b) swap(A, i, j) to swap the ith element and the jth of A.

6. [10 points]

   Prove by induction that: if $n^3 + 2n$, is divisible by 3 for all $n \geq 0$.

7. [10 points]
   Indicate for each statement if it is True or False :
   (a) [2 points] $2n^2 + 1 = O(n^2)$
   (b) [2 points] $\sqrt{n} = O(\log n)$
   (c) [2 points] $\sqrt{n} \log n = O(n)$
   (d) [2 points] $3n^2 + \sqrt{n} = O(n^2)$
   (e) [2 points] $\log n = O(n^{-1/3})$
8. [10 points]
   Programs A and B were analyzed, for A the worst case is less than $150n \log n$ and for B is less than n^2. Answer the questions:
   (a) Which program is faster when $(n > 10000)$?
   (b) Which program is faster when $(n < 100)$?
   (c) Is the program B faster than program$A$ for input of any size?
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 − 20 % 3.0 / 18 / 12) + 7 ∗ 2 / 5;
   (b) **double** b = (**double**) (2.3f + 5) − 20 + 3.4 / (18.1f / 12);
   (c) ffloat c = (2.3f + 5) − 20 / 3 ∗ 18 / 12;
   (d) **float** d = (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) || !((33 − 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.