Exam 1

Data Structures
Thursday, October 7, 2021

You must justify all your answers to recieve full credit

---

1. Consider a code fragment using the bitwise XOR.

```
1 int a = -3;
2 int x = ???  // replace this ??? with a number
3 int b = a^x;
4 cout << hex << b;
5 // This outputs the hexadecimal representation of "b": "ffffabcd".
```

   - Write the hexadecimal representation of variable "a", if its decimal value is -3.
   - Write the binary representation of the same variable "a".
   - Write the hexadecimal representation of variable "x" — how to rewrite Line 2 of the code snippet above to make variable "b" equal to `0xffffabcd`.

2. A doubly linked list has 3 nodes of type `Node`, see Figure 1: it has `prev` and `next` pointers of type `Node*` and also a constant `info` field storing constant positive integers (i.e. `info` field does not change over the lifetime of the given node).

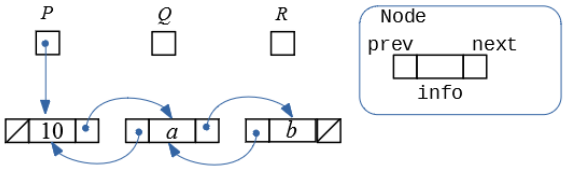

Figure 1: Doubly linked list

   Variables `P, Q, R` are pointers of type `Node*`, initially `P` points to the first node in this list. The `info` fields in this list have values 10, a and b.

   Write a code that modifies this doubly linked list: If $a > b$ it changes the pointers in the list so that the node with info 10 points to b, then b points to a. If $a \leq b$ then your code should leave the list unchanged.

3. Define the following functions $f_1(n)$, $f_2(n)$, $f_3(n)$ mapping positive integers $n \geq 5$ to positive real numbers $t > 0$:

$$\begin{cases} f_1(n) = (1 + \cos n)\sqrt{2^{7 \cdot \log_2 n}}, \\ f_2(n) = 13^{\log_2 n}, \\ f_3(n) = \frac{1}{n^2} \cdot \binom{n}{5}, \\ f_4(n) = f_1(n) + f_2(n) + f_3(n). \end{cases}$$

   For each function find Big-O notation $O(g(n))$, Big-Omega notation $\Omega(g(n))$, and Big-Theta notation $\Theta(g(n))$. Show algebraic and/or verbal justifications to your answers.

4. The following C++ program declares a class `Pair`. Pairs are *lexicographically ordered*: $(x_1, y_1) < (x_2, y_2)$ iff $x_1 < x_2 \ \vee \ (x_1 = x_2 \ \wedge \ y_1 < y_2)$.

   Complete the program to input a positive integer n, then input n pairs from the standard input, push them on the stack while skipping those pairs which are not lexicographically larger than the current top element of the stack. Finally, output the remaining pairs to the standard output as separate lines in their original order.

   Use the overloaded operators "`cout << pair`", "`cin >> pair`", "`p1 < p2`" for input, output and comparisons. The only data structure to use is STL stack. If necessary, you can use several stacks.

```
#include <iostream>
#include <stack>
using namespace std;
class Pair { public: int x, int y };
istream &operator>>(istream  &input, Pair &p ) {
  input >> p.x >> p.y;   return input;
}
ostream &operator<<(ostream &output, const Pair &p ) {
  output << "(" << p.x << "," << p.y << ")";   return output;
}
bool operator<(const Pair &left, const Pair &right) {
  // implement the lexicographic comparison operator.
}
int main() {
  // Input the total number of pairs, then 2*n integers (the pairs).
  // Output those pairs which are in lexicographically increasing order.
}
```

Sample input

```
5
4 17
4 17
5 1000
7 12
7 9
```

Sample output

```
4 17
5 1000
7 12
```

5. In your code from the previous exercise, define the default (no argument) constructor, copy constructor and destructor. Describe the order in which these are called as you run `main()` on this input:

```
2
2 3
1 4
```