CMSC 231: Discrete Mathematics
Practice Exam #2
Fall 2017

- This review exam has *not* been tested for length. The real exam might be shorter or longer.
- The actual exam will have 90 points of standard-difficulty problems and 15 points of challenge problems.
- You will be able to bring one standard sheet of US-letter-sized paper with notes. The notes may appear on both sides of the sheet.
- No calculators or other electronic devices will be allowed on the exam.
- You must give only *one* answer to each question. Attempting to give two different answers to one question will earn no credit.

Definition: If n is an integer, then n is *even* iff:

$$\exists(k \in \mathbb{Z}), n = 2k$$

Definition: If n is an integer, then n is *odd* iff:

$$\exists(k \in \mathbb{Z}), n = 2k + 1$$

1. Prove that if m is an even integer, then $m + 7$ is odd. Do this proof in two ways: proof by contraposition and proof by contradiction.

Definition: If n and d are integers and $d \neq 0$, then we say that d divides n (or $d \mid n$) iff:

$$\exists(k \in \mathbb{Z}), n = dk$$

We can write $d \nmid n$ to mean $\sim(d \mid n)$.

2. Using proof by contradiction, prove: $\forall n \in \mathbb{Z}, 4 \nmid (n^2 + 2)$.

3. (a) Write using summation notation (Σ): $(x^2+1)+(x^2+1)^2+(x^2+1)^3+\cdots+(x^2+1)^n$.

Answer: ____

(b) Write using product notation (Π): $1 \cdot 2^2 \cdot 3^3 \cdot 4^4 \cdot \cdots \cdot k^k$.

Answer: ____

4. Using induction, prove that for all integers $n \geq 1$, $2^{2n} - 1$ is divisible by 3. In other words, prove: $\forall (n \in \mathbb{Z}), (n \geq 1) \rightarrow 3 \mid (2^{2n} - 1)$.

5. Using induction, prove:

$$\forall(n \in \mathbb{Z}), (n \geq 0) \rightarrow \sum_{i=0}^{n} 3^i = \frac{3^{n+1} - 1}{2}$$

Loop Invariant Theorem: Let a `while` loop with guard G be given, together with pre- and post-conditions that are predicates in the algorithm variables. Also let a predicate $I(n)$, called the *loop invariant*, be given. If the following four properties are true, then the loop is correct with respect to its pre- and post-conditions.

I. **Basis Property:** The pre-condtion for the loop implies that $I(0)$ is true before the first iteration of the loop.

II. **Inductive Property:** For all integers $k \geq 0$, if the guard G and the loop invariant $I(k)$ are both true before an iteration of the loop, then $I(k+1)$ is true after the iteration of the loop.

III. **Eventual Falsity of Guard:** After a finite number of iterations of the loop, the guard G becomes false.

IV. **Correctness of the Post-Consition:** If N is the least number of iterations after which G is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.

6. Use the Loop Invariant Theorem to prove that the algorithm below is correct with respect to its pre- and post-conditions.

   Pre-conditions:

   (1) $\texttt{s}$ is an integer such that $\texttt{s} > 0$.

   (2) $\texttt{A}$ is an array of integers with indices $1, \ldots, \texttt{s}$.

   (3) $\texttt{i}$ is an integer such that $\texttt{i} = 1$.

   (4) $\texttt{m}$ is an integer such that $\texttt{m} = \texttt{A[1]}$.

   (5) $\texttt{r}$ is an integer.

```
while (i <= s)
  if A[i] > m, then:
    set r := i
    set m := A[i]
  set i := i + 1
```

   Post-conditions:

   (1) $1 \leq \texttt{r} \leq \texttt{s}$

   (2) $\texttt{A[r]}$ is the maximum integer in the array $\texttt{A}$.

   Loop invariants: $I(n) =$

   (1) $\texttt{i} = n + 1$

   (2) $\texttt{m}$ is the maximum element in the range $\texttt{A[1]}, \ldots, \texttt{A[i]}$

   (3) $\texttt{A[r]} = \texttt{m}$

Write your proof here:

7. (**Challenge problem**) The recursive `mergesort` algorithm takes three inputs: an array `A` of integers, an integer `lo` containing the lowest index (inclusive) in the array which should be sorted, and an integer `hi` containing the highest index (inclusive) in the array which should be sorted. Here is its definition:

```
mergesort(A, lo, hi):
  if hi > lo, then:
    let mid = floor((lo + hi) / 2)
    mergesort(A, lo, mid)
    mergesort(A, mid+1, hi)
    merge(A, lo, mid, hi)
```

The algorithm refers to two functions. The `floor` function rounds a number down to the nearest integer. In other words, `floor(n)` $= \lfloor$`n`$\rfloor$. The `merge` function merges two sorted arrays into one bigger sorted array; its implementation is not important for this problem.

Let s_n denote the number of steps it takes to run —mergesort(A, lo, hi)—, where $n =$ `hi` $-$ `lo`. Here, we can consider one step to be performed by one line of code. Let m_n be the number of steps it takes to run `merge(A, lo, mid, hi)`, where $n =$ `hi` $-$ `lo`.

Write a recursive definition of s_n in terms of m_n. (You do *not* have to define m_n.) Write down any assumptions you have had to make in order to write down this definition.

8. Define a set S recursively as follows:

   I. BASE: $\mathtt{a} \in S$, $\mathtt{b} \in S$

   II. RECURSION: If $s_1 \in S$, $s_2 \in S$, and $s_3 \in S$, then

   $$s_1 s_2 s_3 \in S$$

   III. RESTRICTION: Nothing is in S other than objects defined in I and II above.

   Use structural induction to prove that every string in S has an odd number of letters.

9. Given sets A, B, and C in the same universe U, determine if each of the following statement is true or false. If it is true, then prove it. If it is false, then give a counterexample.

   (a) $((C \subseteq A) \wedge (C \subseteq B)) \rightarrow (C \subseteq (A \cup B))$

   (b) $(C \subseteq (A \cup B)) \rightarrow ((C \subseteq A) \wedge (C \subseteq B))$

(c) $A^c \cap (A \cup B) = B - A$

10. Define $f : \mathbb{Z}^{nonneg} \rightarrow \mathbb{Z}^{nonneg}$ such that $f(n)$ is the sum of the digits in the decimal representation of n.

    (a) Is f injective (one-to-one)? If so, prove. If not, provide a counterexample.

    (b) Is f surjective (onto)? If so, prove. If not, provide a counterexample.

11. (**Challenge problem**) Suppose A is a set, and we have $g : \mathbb{Z}^{nonneg} \to A$ and $h : \mathbb{Z}^{nonneg} \to A$. Furthermore, suppose g and h are both bijections (= one-to-one correspondences = injective and surjective = one-to-one and onto). Define $f : \mathbb{Z} \to A$ as follows:

$$f(n) = \begin{cases} g(n) & n \geq 0 \\ h(-n) & n < 0 \end{cases}$$

(a) Is f injective (one-to-one)? If so, prove. If not, provide a counterexample.

(b) Is f surjective (onto)? If so, prove. If not, provide a counterexample.