

NAME: _____

ID: _____

C241 Test Two

INSTRUCTIONS:

- Put your name and ID number in the upper-right of each page.
- Exam time is 75 minutes.
- There five equally weighted questions.
- Place your final answer on the test in the space provided. Scratch work is not graded, but neatness counts. Mark the parts of your answers clearly.

1	/20
2	/20
3	/20
4	/20
5	/20
	/100

1. (20 points) Use induction to prove that the sum of the first n odd numbers is equal to n^2 . That is, show:

For all natural numbers $n \geq 1$, $\sum_{i=1}^n (2i - 1) = n^2$.

BASE CASE. $\sum_{i=1}^1 (2i - 1) = 2 \cdot 1 - 1 = 1 = 1^2$

INDUCTION. Assume that $\sum_{i=1}^k (2i - 1) = k^2$.

$$\sum_{i=1}^{k+1} (2i - 1) = 2(k+1) - 1 + \boxed{\sum_{i=1}^k (2i - 1)} \quad (\text{expanding } \sum \text{ by one term})$$

$$\stackrel{H}{=} 2(k+1) - 1 + \boxed{k^2} \quad (I.H.)$$

$$= k^2 + 2k + 1 \quad (\text{simplifying})$$

$$= (k+1)^2 \quad (\text{factoring})$$

2. (20 points) *Prove:* If $f(n) \in O(g(n))$ then $[f(n)]^3 \in O([g(n)]^3)$.

If $f(n) \in O(g(n))$ then there exist numbers N and C such that for any number $n \geq N$, $f(n) \leq C \cdot g(n)$.
Hence

$$[f(n)]^3 \leq [C \cdot g(n)]^3 = C^3 \cdot [g(n)]^3$$

Thus, we have shown that $[f(n)]^3 \in O([g(n)]^3)$ with witnesses N and C^3 .

COMMENT. Perhaps one should consider, if only briefly, the cases where $f(n)$, $g(n)$, or C are less than 1. In these cases, $[f(n)]^3$, $[g(n)]^3$, and C^3 get smaller, but the inequalities still hold. Recall also that Definition 5.3 requires that the values be non-negative.

3. (20 points) If A , B , and C are countable (possibly infinite) sets, prove that:

By the definition of countable (Defn. 5.2) there exist injective functions $f_A: A \rightarrow \mathbb{N}$, $f_B: B \rightarrow \mathbb{N}$, and $f_C: C \rightarrow \mathbb{N}$. The answers below use these functions.

- (a) $B \cap C$ is countable.

To show that $S = B \cap C$ is countable, we can use either f_B or f_C to index the elements, but we should “restrict” their domains to the intersection. Define $f_S: (B \cap C) \rightarrow \mathbb{N}$ to be

$$f_S(x) = f_B(x) \text{ for any } x \in B \cap C$$

COMMENT. We should argue that f_S is injective (one-to-one) to satisfy Definition 5.2. This is not required for the test answer, but check yourself by writing out the explanation.

- (b) $A \cup (B \cap C)$ is countable.

To show that $S = A \cup (B \cap C)$ is countable, define $f_R: S \rightarrow \mathbb{N}$ to be

$$f_R(x) = \begin{cases} 2 \cdot f_A(x) & \text{if } x \in A \setminus (B \cap C) \\ 1 + 2 \cdot f_S(x) & \text{if } x \in (B \cap C) \end{cases}$$

where f_S is as defined in Part (a).

f_R is well defined (i.e. a function) because $A \setminus (B \cap C)$ is disjoint from $B \cap C$. Hence there is no possibility of mapping an element x to two distinct values. f_R is an injection because f_A and f_S are, and because elements of $A \setminus (B \cap C)$ are mapped to even numbers while elements of $B \cap C$ are mapped to odd numbers.

- (c) Bonus: $A \times B \times C = \{(a, b, c) \mid a \in A, b \in B, c \in C\}$ is countable

COMMENT. This is an illuminating problem to solve, so no solution is provided.

4. (20 points) This three-part question has to do with program performance analysis. For full credit, give your answers in a form that does not include summation notation (\sum).

Suppose throughout that

program statement S_1 takes 5 units of effort,

program statement S_2 takes 3 units of effort,

conditional test T takes 1 unit of effort, and

assignments, comparisons, and increments each take 1 unit of effort

- (a) For fixed value $N \in \mathbb{N}$, give the worst-case performance estimate for the statement below. That is, what is the *maximum* effort the statement might take?

```
for (i := 5; i < N; i := i + 1)
  if T then S1 else S2
```

(i) The index assignment $i := 5$ is done just once, taking 1 unit of effort. (ii) Each cycle through the loop, $i := i + 1$, $i < N$ and test T together consume 4 units of effort. (iii) In the worst case, the conditional test T always succeeds and S_1 is executed, needing 5 units of effort. (iv) When the loop terminates, $i < N$ is done one last time, taking 1 unit of effort. So the worst-case performance estimate is

$$1 + \sum_{i=5}^{N-1} [4 + 5] + 1 = 2 + \sum_{i=5}^{N-1} [9] = 2 + ((N-1) - 5 + 1) \cdot 9 = \boxed{9N - 43}$$

for $N \geq 4$ of course. If $N < 4$, 2 units of effort are needed to initialize i and compare it to 5. But we are interested in "large N ."

- (b) For the same program, give best-case performance estimate. That is, what is the *minimum* effort the statement might take?

```
for (i := 5; i < N; i := i + 1)
  if T then S1 else S2
```

The loop executes S_2 rather than S_1 taking 2 fewer units of effort. By the same analysis, the performance estimate is $\boxed{7N - 33}$

(c) Give the performance estimate for the nested loop

```
for (i := 1; i < N; i := i + 1)
  for (j := i; j < N; j := j + 1)
    S1
```

The inner loop performs $2 + \sum_{j=i}^{N-1} [4 + 5]$ units of effort. This simplifies to $2 + 9(N - i)$. Let $E(i)$ denote this value. The outer loop takes $2 + \sum_{i=1}^{N-1} E(i)$ units of effort. To eliminate the \sum notation,

$$\begin{aligned}
 \sum_{i=1}^{N-1} E(i) &= \sum_{i=1}^{N-1} [2 + 9(N - i)] && \text{(expanding } E(i)) \\
 &= \sum_{i=1}^{N-1} 2 + 9 \left[\sum_{i=1}^{N-1} N \right] - 9 \left[\sum_{i=1}^{N-1} i \right] && \left(\begin{array}{l} \sum(x+y) = \sum x + \sum y \\ \sum C \cdot x = C \cdot \sum x \end{array} \right) \\
 &= 2(N - 1) + 9(N - 1)N - 9 \left[\sum_{i=1}^{N-1} i \right] && \text{(summation of constants)} \\
 &= 9N^2 - 7N - 2 - 9 \left[\sum_{i=1}^{N-1} i \right] && \text{(simplifying)} \\
 &= 9N^2 - 7N - 2 - 9 \left[\frac{N(N - 1)}{2} \right] && \left(\sum_n^m i = \frac{1}{2}(m - n + 1)(m + n) \right)^\dagger \\
 &= \frac{18N^2 - 14N - 4 - 9N^2 + 9N}{2} && \text{(combining terms)} \\
 &= \frac{9N^2 - 5N - 4}{2} && \text{(simplifying)}
 \end{aligned}$$

Adding 2 (in the form of $\frac{4}{2}$) for index initialization and the terminal test, we arrive at the estimate of

$$\boxed{\frac{9}{2}N^2 - \frac{5}{2}N}$$

The answer is of the right order, since we have seen several examples now of this nested-looping pattern, and all of them have been $O(n^2)$.

[†] This is a generalization of the first example of induction (4.1, p. 48) in the textbook: $\sum_{i=0}^n i = \frac{1}{2}N(N + 1)$. You aren't expected to come up with this formula in a test situation, but doing so demonstrates insight. See the REMARK at the end of the example.

5. (20 points) The postcondition in the program below claims it performs multiplication. Assume that all constants, program variables, conditions, and operations range over natural numbers. (15 points) Carefully state three conditions that suffice to verify this program. (5 points) Prove these conditions.

```

{x = A ∧ y = B}
z := 0;
while y ≠ 0 {z + x · y = A · B}
  if even?(y)
  then begin y := y ÷ 2 ; x := 2 · x end
  else begin y := y - 1 ; z := z + x end
{z = A · B}

```

(a) CONDITION:

PROOF:

(b) CONDITION:

PROOF:

(c) CONDITION:

PROOF:

There are three things to prove

(a) INITIALIZATION: $\{x = A \wedge y = B\} z := 0 \{z + x \cdot y = A \cdot B\}$

If $x = A$ and $y = B$ to begin with, and z is set to 0, then $z + x \cdot y = 0 + A \cdot B = A \cdot B$

(b) INVARIANCE: $\{z + x \cdot y = A \cdot B\} \wedge (y \neq 0) \{ \text{if } \text{even?}(y) \text{ then } \dots \text{else } \dots \{z + x \cdot y = A \cdot B\}$

Assume that $z + x \cdot y = A \cdot B$ and $y \neq 0$. There are two cases to consider,

*CASE A. If y is even, then $y = 2q$ for some $q \neq 0$. The **then** branch computes $x' = 2x$ and $y' = y$, leaving z unchanged. So $z' + x' \cdot y' = z + 2x \cdot p = z + x \cdot 2p = z + xy$*

*CASE B. Otherwise, if y is not even, the **else**-branch computes $y' = y - 1$, $z' = z + x$, and $x' = x$.*

$$z' + x' \cdot y' = (z + x) + x \cdot (y - 1) = (z + x) + xy - 1 \cdot x = z + x + xy - x = z + xy$$

Thus, in both cases we have shown that the loop body computes $z' + x' \cdot y' = z + x \cdot y$ which, by assumption equals $A \cdot B$ as desired.

(c) TERMINATION: $(z + x \cdot y = A \cdot B) \wedge (y = 0) \Rightarrow (z = A \cdot B)$

If $z + x \cdot y = A \cdot B$ and $y = 0$, then $z + xy = z + 0 = z = A \cdot B$, so the postcondition holds.