UNIVERSITY OF TORONTO

Faculty of Arts and Science

term test #2 SOLUTIONS
CSC236

Date: Monday March 23, 2020
Duration: 50 minutes
Instructor(s): Colin Morris

Please read the following guidelines carefully!

- This test is open book. You may consult your notes, along with any published course materials. Do not consult with anything or anyone else.
- If you would like clarification about a question, you can post it as a **private** question on Piazza, and I will answer it promptly.
- After 50 minutes, you must stop writing and upload your solutions to MarkUs. (I recommend setting a timer to avoid losing track of time.)
- You have **10 minutes** to upload your solutions to MarkUs (or 20 minutes if you are uploading handwritten solutions).
- We will accept just about any format, though a pdf is preferred. Your first priority should be getting your solutions into MarkUs *in some form* by the deadline. If you need to make superficial revisions afterward (e.g. rotating or cropping images) this is acceptable.
- More in-depth logistical details are available on the course website
- This examination has **3** questions. There are a total of **4 pages, DOUBLE-SIDED**.

Good luck!

1. **[9 marks]** (≈ 20 minutes) Recall the following generalized form for divide-and-conquer recurrences:

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ f(n) + aT(n/b) & \text{if } n > 1 \end{cases}$$

(a) Let $T_0(n)$ be a generic divide-and-conquer recurrence having $f(n) = 0$. i.e. in the (unrealistic) scenario where there is no non-recursive work associated with each invocation. Use unwinding to devise a closed form for $T_0(n)$. Your closed form should involve the variables a and b. You may assume n is a power of b.

Solution

$$\begin{aligned} T_0(n) &= aT_0(n/b) \\ &= a(aT_0(n/b)) \\ &= a(a(aT_0(n/b))) \\ &\dots \\ &= a^k T_0(1) \\ &= a^k \end{aligned}$$

Where k is the number of times I must divide n by b before reaching 1. By definition, this means $k = \log_b n$. So my closed form is $T_0(n) = a^{\log_b n}$.

(b) Let $T_1(n)$ be a generic divide-and-conquer recurrence having $f(n) = 1$. i.e. $T_1(n) = 1 + aT_1(n/b)$. Find a specific pair of values for $a, b \in \mathbb{N}$ ($a > 0$, $b > 1$) where $\Theta(T_0(n)) = \Theta(T_1(n))$. Briefly justify your answer. (To be clear, you should fix the same a, b values for $T_0(n)$ and $T_1(n)$.) You may use the Master Theorem.

Solution Let $a = 2, b = 2$. Since $T_1(n)$ has $f(n) = 1 = n^0$, we have $d = 0$, and $a > b^d$. So the Master Theorem says that $T_1(n) \in \Theta(n^{\log_b a}) = \Theta(n)$.
$T_0(n) = 2^{\log_2 n} = n$, so $T_0(n)$ is also in $\Theta(n)$.

(c) Repeat the above, finding values for a, b such that $\Theta(T_0(n)) \neq \Theta(T_1(n))$.

Solution Let $a = 1, b = 2$. Now we have $a = 1 = b^d$, so the Master Theorem says $T_1(n) \in \Theta(n^d \log_b n) = \Theta(n^0 \log_2 n) = \Theta(\log n)$.
$T_0(n) = a^{\log_b n} = 1^{\log_2 n} = 1$. So $T_0(n)$ is constant, whereas $T_1(n)$ is logarithmic.

2. **[5 marks]** (≈ 10 minutes) Consider the following function `isuniform`, and the subsequent "proof" of its correctness.

```
1  def isuniform(A):
2      """Pre: A is a list
3      Post: return True iff every element in A is the same
4      """
5      if len(A) <= 1:
6          return True
7      m = len(A) // 2
8      L = A[:m]
9      R = A[m:]
10     return isuniform(L) and isuniform(R)
```

1 I will prove the correctness of `isuniform` by complete induction on the size of the input.
2 Let $n \in \mathbb{N}$. Assume that the function is correct on all inputs of size smaller than n.
3 Let A be a list of size n satisfying the precondition. I will prove that the function is correct on input A.
4 By properties of integer division (as shown in lecture), $\text{len}(L) = \lfloor n/2 \rfloor$, and $\text{len}(R) = \lceil n/2 \rceil$.
5 Thus, L and R are non-empty sublists which partition A.
6 Since they are smaller than A (and satisfy the precondition), by the inductive hypothesis, each of the recursive
7 calls returns True iff the corresponding sublist is uniform.
8 Case 1: A is uniform. Then sublists L and R must also be uniform. So by the IH, line 10 returns True, as required.
9 Case 2: A is not uniform. Then, by definition, there must be some index i such that $A[i]$ is different from the other
10 elements.
11 If $i < m$, then that element is part of sublist L, therefore L is not uniform, and `isuniform(L)` returns False.
12 Otherwise, $i \geq m$, so the differing element is part of sublist R, therefore R is not uniform, and `isuniform(R)`
13 returns False. Either way, we return False from line 10, as required.
14 In both cases, `isuniform` terminates and satisfies the postcondition for an arbitrary list A of size n.
15 Therefore, by complete induction, the function is correct on inputs of all sizes.

Identify the logical flaw(s) in this proof. The lines of the proof have been numbered so you can more easily refer to them. **Note:** You are *not* being asked to find a bug in the code of `isuniform`. Your answer should instead refer to why the proof above is not valid.

Solution There are two problems with this proof:

(a) It lacks a base case, in the sense that the proof needs to separately deal with the case where $n \leq 1$. Line 4 of the proof implicitly assumes that the program reaches line 7 of the code (and beyond), which is not the case when $n \leq 1$.

(b) In Case 2, it is true there must exist an i such that $A[i]$ differs from *some* other element j[1]. But it does not follow that the demi-list containing index i is non-uniform, because the only differing index j might be in the *other* half of A. e.g. if $A = [1, 2]$, then the halves L and R are both uniform, though A itself is not.

[1] in fact, this is true of all valid indices i

3. **[8 marks]** (≈ 20 minutes)

Consider the following code

```
1  def pal(s):
2      """Pre: s is a string
3      Post: return True iff s is a palindrome, i.e. s == s[::-1]
4      """
5      a = 0
6      b = len(s) - 1
7      r = True
8      while a < b:
9          r = (r and (s[a] == s[b]))
10         a += 1
11         b -= 1
12     return r
```

(a) Indicate which of the below statements are valid loop invariants (i.e. which are true at the end of every iteration j of pal's loop). For example, if you think the true invariants are the first and fourth ones from the first row, all of the second row, none of the third row, and the first invariant from the last row, then your answer should look like:

1) A D
2) A B C
3)
4) A

(If you are annotating this test paper directly, you may also indicate your selections by circling or underlining the true invariants.)

(1) (A) $a_j < b_j$ (B) $a_j \leq b_j$ (C) ✓ $a_j \leq b_j + 1$ (D) $a_j > b_j$ (E) $a_j \geq b_j$
(2) (A) ✓ $a_j \in \mathbb{N}$ (B) $b_j \in \mathbb{N}$ (C) ✓ $r_j \in \{True, False\}$
(3) (A) $s[a_j : b_j]$ is a palindrome (B) $s[a_j : b_j]$ is a palindrome iff r_j is True
(4) (A) ✓ s is a palindrome ⇒ r_j is True (B) r_j is True ⇒ s is a palindrome

Correct invariants are marked above with a ✓.

Marking note: Each row above was assigned 1 point, and was marked on an all-or-nothing basis.

(b) Show that pal terminates by demonstrating a loop measure m_j, such that for every iteration j

- $m_j \in \mathbb{N}$
- $m_j < m_{j-1}$ (for $j \neq 0$)

You do not need to prove or explain why your m_j has these properties.

Solution There are many correct answers to this problem. Below are a few examples:

i. $m_j = b_j + 10$
ii. $m_j = \text{len}(s) - a_j$
iii. $m_j = (b_j - a_j) + 10$

Note that the first and third example won't work without an added constant. (10 is larger than necessary, but we might as well use a generous constant to save ourselves any anxiety about off-by-one errors.)

We would also (begrudgingly) accept a loop measure such as $m_j = a_j - j$.

(c) Prove **one** invariant (of your choice) from part (a).

Solution The easiest invariants to prove are the ones from row (2). We present here a proof of invariant 2 A): $a_j \in \mathbb{N}$.

Base case: $a_0 = 0 \in \mathbb{N}$ (by line 5)

Assume $a_j \in \mathbb{N}$ for some j. Assume a $j+1$th iteration. By line 10, $a_{j+1} = a_j + 1$. The naturals are closed under addition, so $a_{j+1} \in \mathbb{N}$.