CPSC 121: Models of Computation
Quiz #2: Section **SAMPLE**, 2009 March 18/19

Name: _____ Student ID: _____

Signature: _____

- The cover page on the real quiz will be identical to this one except that it will not include this line and will indicate the appropriate section and date above. **Read these instructions now!**
- You have **30 minutes** to write the 3 questions on this quiz.
- A total of **16 marks** are available. You may want to complete what you consider to be the easiest questions first!
- Ensure that you clearly indicate a single legible answer for each question.
- You are allowed a single 8.5" x 11" reference sheet. The sheet must have your name on it and may contain any content you like. Otherwise, no notes, aides, or electronic equipment are allowed.
- Good luck!

UNIVERSITY REGULATIONS

1. Each candidate must be prepared to produce, upon request, a UBCcard for identification.

2. Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.

3. No candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination.

4. Candidates suspected of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action:

   - having at the place of writing any books, papers or memoranda, calculators, computers, sound or image players/recorders/transmitters (including telephones), or other memory aid devices, other than those authorized by the examiners;
   - speaking or communicating with other candidates; and
   - purposely exposing written papers to the view of other candidates or imaging devices. The plea of accident or forgetfulness shall not be received.

5. Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

6. Candidates must follow any additional examination rules or directions communicated by the instructor or invigilator.

Notes about this Sample Quiz

Along with the *many* other practice resources available to you in the textbook and on the course website, this sample quiz is intended to prepare you for the upcoming in-class quiz on March 18 or 19, 2009.

Although the in-class quiz will of course differ from this sample, its structure and some of the features of its questions will also appear on the actual quiz. We note guaranteed similarities in footnotes on each question.

1 Creating Induction Proofs[1] [6 marks]

Part of the marks for this question will be given for writing the framework of the proof clearly and completely. In other words, you will be marked on the form of your proof as well as on its content.

Prove by induction on n that for every positive integer n, $\sum_{i=1}^{n} \frac{1}{i^2} \leq (2 - \frac{1}{n})$.

(Aside: You don't need the following information, but it is interesting to note that $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$.)

[1] The in-class quiz will contain an induction proof problem; the first paragraph will also be the same.

2 Critiquing Induction Proofs[2] [5 marks]

Consider the following induction proof. The proof contains one or more flaws; however, the theorem to be proved is true and provable by induction. Your job is to create a correct proof. The easiest way to do so is likely to be by crossing out and correcting erroneous lines.

Note: the definitions and statement of the theorem are *not* flawed.

Definitions: Consider the following (correct) algorithm to find the maximum element in a list of integers.

FindMax:

1. If the list is of length 1, return the only element.
2. Otherwise, divide the list (as nearly as possible) in half.
   (a) Use FindMax to find the largest element in the right half; call it max_r
   (b) Use FindMax to find the largest element in the left half; call it max_l
   (c) Compare max_r and max_l; if max_r is larger, return it; else, return max_l.

Theorem: for all positive integers n, for any list of integers of length n, FindMax takes $n - 1$ comparisons to find the maximum element in the list.

Proof: We proceed by induction.

Base Case: A list of length 0 has no maximum element. ✓

Induction Hypothesis: Assume for an arbitrary integer $k \geq 1$ that it takes $k - 1$ comparisons to find the maximum element in a list of length k.

Inductive Step: Given the Induction Hypothesis, we must show that it takes $(k + 1) - 1$ comparisons to find the maximum element in a list of length $k + 1$.

Since $k + 1 \geq 2$, we always divide the list "in half" and find the maximum of each half.

Since $k + 1 \geq 2$, $(k + 1)/2 \geq 1$. Therefore, $\lfloor (k + 1)/2 \rfloor \geq 1$ and $\lceil (k + 1)/2 \rceil \geq 1$, and both halves contain at least one element. Since both halves contain at least one element, neither half is as long as $k + 1$ (or their total length would be greater than $k + 1$).

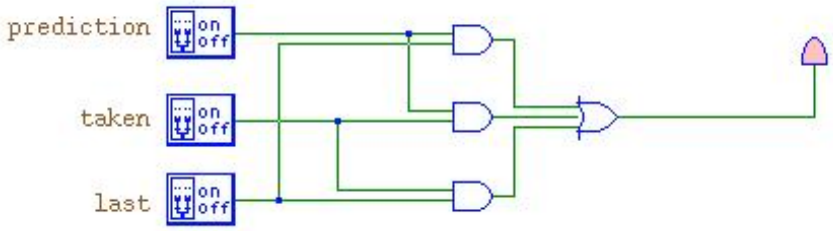
Therefore, the Induction Hypothesis applies to lists of length $\lfloor (k + 1)/2 \rfloor$ and $\lceil (k + 1)/2 \rceil$.

So, a list of length $k + 1$ takes $\lfloor (k + 1)/2 \rfloor + \lceil (k + 1)/2 \rceil = k + 1 \leq (k + 1) - 1$ comparisons. □

[2] The quiz will have a problem with the same structure and instructions—an induction proof critique—but a different proof with (perhaps) a different number and different types of flaws.

3 Sequential Circuits[3] [5 marks]

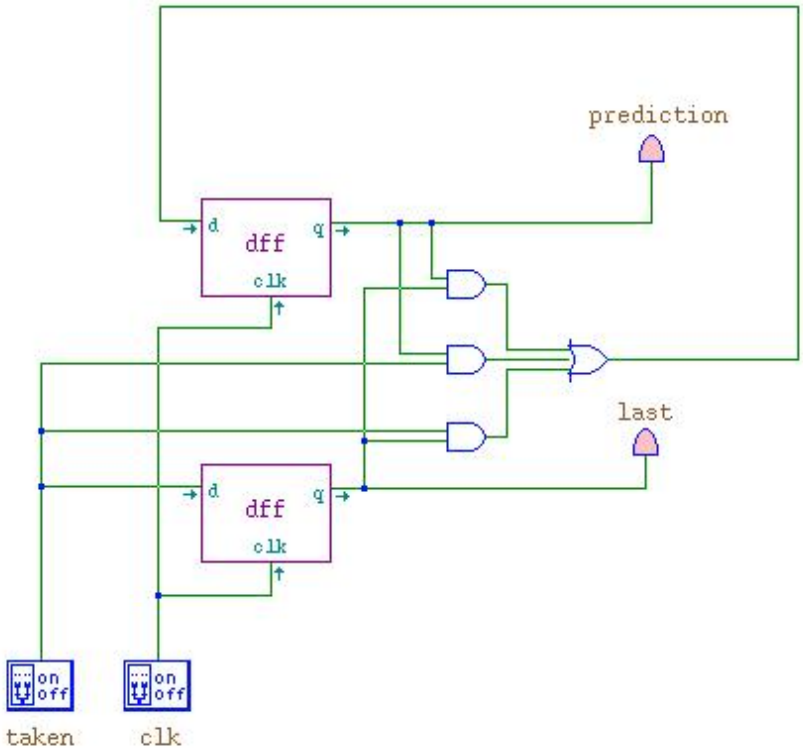
1. Fill in the following truth table for the combinational circuit shown. (Note that this is the key combinational component of the sequential circuit below.)



prediction	*taken*	*last*	*output*
0	**0**	**0**	
0	**0**	**1**	
0	**1**	**0**	
0	**1**	**1**	
1	**0**	**0**	
1	**0**	**1**	
1	**1**	**0**	
1	**1**	**1**	

[3]The sequential circuits question on the quiz will have exactly the same structure as this question although a different circuit, truth table, and list of inputs.

2. The table below shows the sequence of input values (i.e., values of $taken$) to the following sequential circuit each time the clock ticks (goes from low to high). Fill in the appropriate value for $prediction$ and $last$ immediately **after** the clock ticks for each row. Use 1 for high/true, 0 for low/false, and U for undefined (yellow in TkGate).[4]



$taken$	$prediction$	$last$
0		
0		
0		
1		
0		
1		
1		
0		
1		
1		
0		
0		

[4]This circuit implements a DFA used to perform branch prediction in computers; essentially, this means predicting whether the then or else branch of a conditional is taken.