Name: ____

CS 620 Fall 2010 at ISU, Exam 2 SAMPLE

Prepared by assistant professor Jeff Kinne on October 25, 2010. You have until 4:00pm to finish the exam. The exam itself and blank paper I provide are all you will have to use (no computer, textbook, notes, cellphone, calculator, etc.). I have put point values on the problems so the total adds up to 24.

Problem 1 (5 points) Recall that PP consists of problems that can be solved by a poly-time randomized machine M such that for every input x, $\Pr_r[M(x,r) \text{ is correct}] > \frac{1}{2}$. Show that SAT is contained in PP. (By the NP-completeness of SAT, this shows that NP is contained in PP.)

Problem 2 (3 points) Recall that RP consists of problems that can be solved by a poly-time randomized machine M such that: (i) for "yes" instances x, for every input x, $\Pr_r[M(x,r) = 1] \geq \frac{1}{2}$, and (ii) for "no" instances x, $\Pr_r[M(x,r) = 1] = 0$. Show that RP is in NP.

Problem 3 (5 points) Recall that ZPP consists of problems that can be solved by a poly-time randomized machine M that can output 0, 1, or ? such that:
(i) for all x, $\Pr_r[M(x,r) \text{ outputs correct answer (0 or 1)}] \geq \frac{1}{2}$, and
(ii), for all x, $\Pr_r[M(x,r) \text{ outputs wrong 0/1 value}] = 0$.

Let M' be a randomized machine that solves a problem Π in the following way. When it outputs a 0/1 value, it is always correct. The running time could be larger than polynomial, but the expected running time is small. That is, if $t_{M(x,\cdot)}$ is a random variable for the running time of M on input x, then $E[t_{M(x,\cdot)}] \leq |x|^c$ for some constant c.

Show that Π is in ZPP. That is, convert M' into a randomized machine M that satisfies the definition of ZPP.

Hint: use Markov's inequality.

Problem 4 (3 points) Show that given a randomized machine M satisfying the definition of ZPP for solving a problem, we can reduce the error to less than $\frac{1}{2^n}$ while maintaining polynomial running time.

Problem 5 (3 points) Let M be a poly-time randomized machine that satisfies the definition of BPP for solving a problem. We want to replace the random bits of this algorithm by the output of a pseudorandom generator G such that the majority vote of running $M(x, G(s))$ for all possible seeds s is correct. How many bits does G need to output? What types of tests does G need to be pseudorandom against?

Problem 6 (5 points) Suppose we roll a fair 6-sided dice 5 times, and assume all rolls are independent of each other. What is the expected value of the sum of the 5 rolls. What is the probability the sum is greater than 27?