QA Week 9

1	**Could we do some further examples on how to do calculations with congruences? I don't know how to calculate for example a = (12 (mod 14)) (mod 14)**	**What you are writing is not really defined. You can say a = 12 (mod 14), which means that a divided b 14 has remaineder 12. Or you can write a = ((12 mod 14) mod 14) = (12 mod 14) = 12 / But that is exactly the form we get in the series at ex.3.1 when we insert 11 (mod 19) for a / Isn't the answer 12? / It's 0, I think because 12 mod 14 = 12 * 1 + 2, so 2 is the remainder, 2 mod 14 = 7*2 + 0 = 0 / no it is 12 because 12 is smaller than 14 / 12 (mod 14) = remainder of 12 by 14 not the other way around**
2	Can we have a mock-exam?	I already mentioned that we plan this for the last two weeks.
3	can you explain the complexity of the algorith that multiply two n_bit integers	If you analyse the algorithm you see that in the second loop, where you add up the partial products, you are performing n additions. Since we have shown that one addition has cost O(n) we obtain as cost for multiplication O(n*n).
4	What is a composite integer ?	nor prime / not* / It's a number that is not prime and he can be written as product of power of primes numbers / It is an integer n that can be divided by an integer that is not 1 and not the integer n itself
5	Have all the properties on prime numbers been discovered or are there some that are still "hidden"	There will be always properties that we do not know about, and there will be properties that we cannot decide (that was the famous result of Gödel)
6	Why the aditive inverse of the arithmetic modulo is m-a and not only -a? since m modulo m is 0	m-a and -a are congruent modulo m but -a is not in the remainder class of m. / Thanks, correct answer
7	is there a method that we can use to write an integer in product of primes	Well you can iterate from 2 to sqrt(n) and find all the prime divider in this area. Then if a is a prime diviser then n/a is also a divider / Oui, euclids algorithm, called "prime factorization", look on google / Indeed searching for factors up to sqrt(n) is the straightforward approach, though expensive. A lot of research tries to improve on this.
8	By “π(x) approches x/ln(x)”, is it mathematically correct to say that lim_(n → ∞) π(x)/(x / ln (x)) = 1?	Yes, that's the definition of asymptotical equivalence. / Yes, this is the right way to express this.
9	But Euclidean Algo breaks our encryption lagorithm? like RSA	No. Computing gcd is different from finding a factorization. Factorization is hard (for non-quantum computers)
10	can you explain binary modular exponentiation	live answered
11	Is it possible to have exercices correction of this week?	Yes, we'll publish them.
12	How should we proceed when we do prime factorisation?	You can search for prime factors by testing all numbers up to sqrt(n). Then you divide yb hte factor, and restart the search. Hoever searching for factors up to sqrt(n) is a straightforward approach, but expensive. A lot of research tries to improve on this.
13	where are the answers… and how can i aces sthe quiz after because i found this one tough	You find the answers always on the github, together wtih the other materials of the week: https://github.com/LSIR/AICC-I/tree/master/Lectures/Week%208 (that was last week)
14	Could you explain the first question with the multiplicative inverse?	'-2*3 mod 7 = -6 mod 7 = 1 mod 7 = 1 / can you explain how you go from -6 mod 7 to 1 mode 7? Is it a property or..? / '-6 mod 7 = -6 + 7 (mod 7) = 1 mod 7 / thanks
15	for the last one i think the 3rd and the 4rth are both correct	No. The 4th answer is an expression using a syntax that we never defined, i.e. it is meaningless.
16	Could you explain question 3 please ?	live answered
17	Can we do a correction because it had no meaning to do this exxercices without eplaining our mistakes	You find the answers always on the github, together wtih the other materials of the week: https://github.com/LSIR/AICC-I/tree/master/Lectures/Week%208 (that was last week) If you have difficulties with an explanation, ask your assistant or post a question on Piazza. / thank you !
18	Hello, it is possible to not put the answer of questions into the questions in the quiz posted on github, but after ? Because if we revise the questions, we have the answer in the potentials answers...	You find the answers always on the github, together wtih the other materials of the week: https://github.com/LSIR/AICC-I/tree/master/Lectures/Week%208 (that was last week)
19	In the definition of lcm(a,b), shouldn't it be the absolute value of a*b instead of a*b?	live answered
20	Quiz 3) why a number > 32 need 6 bits but not 7 bits?? Because with 6 we can only reach 32 (we want numbers that are > 32)	With 6 bits you can get: 11 1111, this is equal to 63 > 32. What happens is that the bit #6 is the one that makes it possible to attain a binary number >= 32. Then you use the ones after the bit #6 to make higher numbers. / The binary representation of 32 has 6 bits, 32 = (100000)_2. The question was about the minimum number of bits any number larger than 32 would require, i.e. min {#bits(n) | n > 32} = 6
21	Could you explain the complexity of the binary modular exponentiation algorithm? Specifically why (log m)^2 is part of it?	Will provide an explanation right away
22	Could you give the explanation right after each question? It would be much more useful.	You find the answers always on the github, together wtih the other materials of the week: https://github.com/LSIR/AICC-I/tree/master/Lectures/Week%208 (that was last week) I prefer to keep it this way, also in the interest of time. But I will try to explain the points that are mentioned in the life stream for some of the questions.
23	Can u explain the formula : ab= gcd(a, b).lcm(a, b)	here's an example (it's a bit hard to explain by writing) : 2^2 +3^4 + 7^3 = A 2^3 + 3^2 + 7^5 = B Then, for the gcd, you take the smaller exponents : 2^2 + 3^2 + 7^3 (which is evidently a diviseur of A and B) For the lcm, you take the grater exponents : 2^3 + 3^4 + 7^5. Then, you easily see that multiplying gcd and lcm is equivalent to multiply a and b, because for the gcd, you took the smaller exponents, for the lcm, you took the greatest, but not both. Multiplying gcd and lcm keeps the uniqueness of prime factorization. / if you're still unsure about it you can ask on BA1 discord / The explanation with taking the smaller and larger exponents of either number for the gcs and lcm is good. Just replace the + by * in the example.

24	i believe that you have the wrong order. 12 (mod14) = remainder of 12 divided by 14	sorry, I meant to comment / live answered
25	why (11)base10 = (1011)base 2	1011 in base 2 = 1*2^3 + 0 *2^2 + 1 *2^1 + 1*2^0 / in base 10 / This is 8 + 2 + 0 + 1 / Thanks for the answer
26	how we find binary representation of a number, can you give exemple please	See slide 37 of week 9. Informally you perform subsequent divisions by the base, and the remainders are the digits of the binary represenation. Simple example: binary expansion of 3: 3 mod 2 = 1; last digit is 1, quotient is 1 1 mod 2 = 1; fist digit is 1, quotient is 0; therefore alogirthm stops binary representation: 3 = (11)_2
27	Is 2 the better base (considering complexity of algorithm ?)	Which algorithm? In general, e.g. for arithmetic operations, the base does not affect complexity. It is like multiplying the cost by some constant.
28	Can we have a definition of a multiplicative mod inverse?	For an integer a in Z_m a multiplicative inverse of a is an integer b in Z_m such that a*_m b = 1 (or a*b mod m = 1)
29	im confused… sometimes u put big O, sometimes u put big theta for cmplexity	Since we are looking at worst time complexity, most of the time we are interested in an upper bound for the complexity and use therefore big-O. But in fact, in many cases we can say more, that is where we can use Theta as well. In practice, often people confuse the two and use big-O when the mean actually Theta.
30	why do the slides always have Big-O for the complexity and not theta?	Since we are looking at worst time complexity, most of the time we are interested in an upper bound for the complexity and use therefore big-O. But in fact, in many cases we can say more, that is where we can use Theta as well. In practice, often people confuse the two and use big-O when the mean actually Theta.