GEMS, FEBRUARY 28, 2009: PROJECT HANDOUT

LENNY FUKSHANSKY

Abstract. This project handout consists of two parts - first some problems on circle packing, and then on binary encoding.

1. Circle packing

Recall that $\pi = 3.14\dots$ is an irrational number that is very important in mathematics. In particular, a circle of radius R has area equal to πR^2, so π is precisely the area of a unit circle (that is, a circle of radius one). Recall also that the area of a square of side-length L is L^2.

If N circles of radius R are packed into a square of side-length L, then the **density** of this packing is defined as

$$D = \frac{\text{Total area of all circles}}{\text{Area of the square}} = \frac{N\pi R^2}{L^2}.$$

When we say that the circles are **packed**, we mean that they do not overlap.

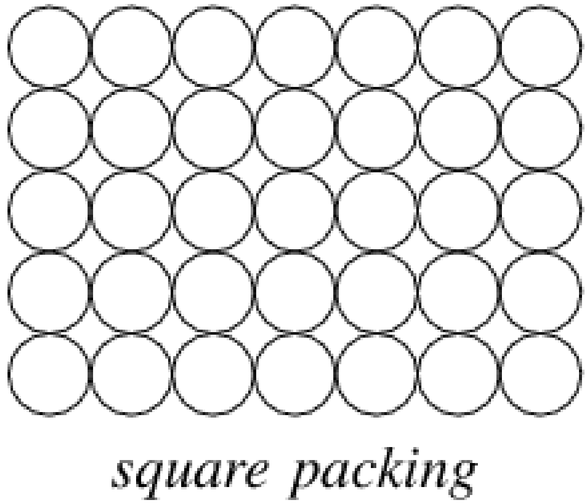
Problem 1. Show that for any packing of circles in a square, $0 < D \leq 1$. Do you think it is possible for $D = 1$? Why or why not?

Problem 2. Do you think it is possible for 16 circles of radius 4 to be packed into a square of side-length $16\sqrt{\pi}$? Why or why not?

Problem 3. Suppose that P_1 is a packing of 10 circles of radius 2 into a square of side-length 20, and P_2 is a packing of 20 unit circles into a square of side-length $10\sqrt{\pi}$. Which packing has higher density? Explain.

Problem 4. If unit circles are packed into a square in such a way that precisely 10 of them touch one side of this square. What can you say about the side-length of this square? Explain.

Problem 5. Suppose that a **square arrangement** of unit circles



is packed into a square of side-length 20. How many circles are there? What is the density of this packing?

2. Binary encoding

Here we talk about a method of encoding information using **binary codes**. Binary arithmetic is based on writing numbers to the base 2 instead of the usual decimal (base 10) system. This means that we are only allowed to use digits 0 and 1 to represent numbers. For example,

$$0 = 0,\ 1 = 1,\ 2 = 10,\ 3 = 11,\ 4 = 100,\ 5 = 101,\ 6 = 110, \dots \quad (1)$$

Problem 1. Continue the examples in (1) to represent all integers up to 26 in binary system.

Problem 2. Can you now formulate a general principle? In other words, let A be a positive integer, written with digits $n_1, \dots, n_k$ in decimal system, that is

$$A = n_1 n_2 n_3 \dots n_k.$$

Can you explain how to write A in binary system?

Problem 3. Now we can use binary system to do coding. Enumerate all the letters A through Z in English alphabet by numbers from 1 to 26, and use your findings from Problem 1 to convert them to binary system; for instance

$$A \to 1 \to 1,\ B \to 2 \to 10,\ C \to 3 \to 11, \dots$$

Continue this list to obtain the binary encoding for the entire alphabet.

Problem 4. Use your conversion table from Problem 3 to write down binary encoding for the sentence "I LOVE MATH".

Problem 5. Suppose I send you an encoded binary message:

11 1111 100 1001 1110 111

1001 10011

110 10101 1110.

Can you decode it into words?

Remark. The binary code is used to store data in a computer, as well as in most other digital devices. Although just a simple binary code as we constructed here does not compress data to allow for fast transmission and does not do error-correction, there are more sophisticated variations of the binary code which do that; some of them also have real connections to sphere packing.

Department of Mathematics, 850 Columbia Avenue, Claremont McKenna College, Claremont, CA 91711

E-mail address: lenny@cmc.edu