Punch Card Sorting
(Companion to Punch Card Searching)

Age group: 11 – adult

Abilities assumed: None

Time: From 10-15 minutes upward
(depending on how detailed the explanations given)

Size of group: 1 upwards

Focus

Sort algorithms
Binary Radix Sorting
Divide and Conquer
Computational Thinking: algorithmic thinking
Practical uses of binary

Syllabus Links

This activity can be used (for example)

- as a general introduction to computing topics from KS3 up.
- to introduce computational thinking problem solving from KS3 up,
- to teach specific syllabus topics such as:
  KS3: understand several key algorithms (sorting) that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem

Summary

Demonstrate how early computers were able to sort data stored on punch cards. It aims to show initially in a magically surprising way how a sort algorithm works. In doing so you demonstrate a practical use of binary numbers, a divide and conquer algorithm for sorting and see an example of a parallel (i.e., where computation is done in parallel) algorithm.

Technical Terms

Sort algorithms, radix sorting, divide and conquer, binary numbers, computational thinking.

Materials

Set of punch cards made from A4 card.
A pencil
Slides, handouts or similar illustrating binary number representations
(see linked slides at www.teachinglondoncomputing.org)

What to do

Set-up:
Create a set of punch cards by printing those given at the end of this booklet. To print without margins you will need to choose as the paper size "borderless A4" if your printer allows it. If not guillotine the margins. Ideally print them on to thin card. Then cut out the notches / holes as indicated by the dotted lines. Sprinkling talcum powder on the cards can help stop them sticking together during the activity (the algorithm will go wrong if the cards stick).

The activity:
Take the pile of punch cards and thoroughly mix them up so they are in a random order. Explain you can get them back into order in no time at all with a little computing magic.

Place a pencil in the right most hole (marked 1) and carefully shake out all the loose cards. Make sure all are shaken out and none stuck in place. Place those cards to the **back** of the pile. Now do the same again but this time placing the pencil through the second hole (marked 2). Shake out the cards and place them at the back of the pile. Do the same again repeatedly moving on a hole at a time and placing the cards at the back.

Challenge the students to explain how it happens. Is it magic?

Binary
To understand how it works you need to know about **binary**. In fact it is a very tangible, physical use of binary. Binary is just a useful way of representing numbers. It is an alternative to the decimal system we use, but where you only have two digits to count with instead of 10 digits. In decimal you count up to 9 before carrying in to the next column to become 10. In binary you count 0,1 before carrying to the next column to get 10. This same pattern continues carrying in to the subsequent column: after 10 you count to 11 then run out of digits so carry over to 100. You count: 0, 1, 10, 11, 100, 101, 110, 111, …

Binary on punch cards
The pattern of holes across the punch cards spell out the binary of the number on the card.

- A **hole** represents a **0**.
- A **slot** cut in to the hole represents a **1**.

So for example 5 in binary is 00101 and the pattern of slots and holes on the card marked 5 (running left to right) is hole-hole-slot-hole-slot, or 0 - 0 - 1 - 0 - 1

Binary Radix Sorting
What we have done is actually to follow a sort algorithm called binary radix sorting. Let's see an example of it in action. Suppose you have cards numbered 0-7 in the order:

3, 5, 6, 1, 2, 0, 4, 7

In binary that is:

011	101	110	001	010	000	100	111
3	5	6	1	2	0	4	7

If we split off those numbers with a 1 in the rightmost position (ie the odd numbers),

011	101	001	111	**and**	110	010	000	100
3	5	1	7		6	2	0	4

If we then move those with a 1 to the back of the stack of cards, keeping the order otherwise the same, we get a new stack as follows:

110	010	000	100	011	101	001	111
6	2	0	4	3	5	1	7

All those bigger in the first digit are now behind those smaller in that digit.

Now doing the same on the middle digit we get:

000	100	101	001	110	010	011	111
0	4	5	1	6	2	3	7

Now those bigger on the second digit are behind those smaller in the second digit. At this point if you ignore the last digit, then based on the rightmost two digits they are in pairs in binary order: 00, 01, 10, 11. We still have to get the last digit right. Finally moving those with a 1 as that leftmost digit to the back we get a sorted stack of cards:

000	001	010	011	100	101	110	111
0	1	2	3	4	5	6	7

We have sorted them in only three steps!

This algorithm is fast for two reasons. Firstly the comparisons are done in parallel, so with one operation (aided by a pencil and gravity) the pile is split in two around the digit being checked. A lot of work is done in each step. This shows the power of such **parallel algorithms**.

The efficiency is more than just that though. The algorithm is using a form of divide and conquer problem solving. Each separate step halves the problem. It leaves a new problem that is essentially the same, just half as big. A sort problem still remains but is smaller in the sense that fewer digits are left to be sorted.

For this algorithm, the pack is divided in half each time based on whether a specific digit is 1 or 0. Each pass sorts the pack up to another digit in the binary number. After one pass the first digits of the cards are in sorted order (0, 1). After two passes a second digit is sorted (00, 01, 10, 11), and so on.

Variations and Extensions

Make a set of punch cards
Have the class make their own set of punch cards, and work out where to cut the slots on each card themselves. Blank punch cards are provided on the website.

Student binary numbers
To demonstrate binary numbers, spell out the numbers on a row of students. This could be done by students holding cards with 1s and 0s as well as the amount a 1 stands for in that position (1,2,4,8, etc). Alternatively have a student standing represent 1 and sitting (curled in a ball) represent 0. See the CS Unplugged binary number activity (www.csunplugged.org) for ideas along these lines of how to introduce binary.

Punch card searching
Challenge the class to come up with a similar algorithm but where cards are discarded that can leave you holding any desired card. (The resulting algorithm is the basis of a separate activity - see below)

Further Reading

The Magic of Computer Science

There are lots more magic tricks with computer science twists available from http://www.cs4fn.org/magic/ including several free magic books.

Links to other activities

The following activities are also available via teachinglondoncomputing.org

The Australian Magician's Dream

Do a magic trick where you predict a card chosen that even the person choosing couldn't have known. Challenge the audience to work out how it is done, teach them how to do the trick and then use it to explain algorithms, searching, and logical reasoning.

Punch Card Searching

Demonstrate how early computers were able to find data stored on punch cards. Show that it is the same algorithm as the magic trick in the 'Australian Magician's Dream' Activity. See in a visual way how a search algorithm works. In doing so you demonstrate a practical use of binary numbers, a divide and conquer algorithm for searching and explore the computational thinking trick of translating solutions between problem areas.

Locked-in

Explore the design of an algorithm to allow someone who is totally paralysed to communicate.
This gives an introduction to computational thinking based problem solving, leading to an understanding of what an algorithm is and how algorithms can be compared on the basis of efficiency. It also illustrates how computational thinking is about more than just creating computer-based solutions. Computing is about solving problems for people.

Live demonstration of this activity

Teaching London Computing give live sessions for teachers demonstrating this and our other activities. See http://teachinglondoncomputing.org/ for details. Videos of some activities are also available or in preparation.

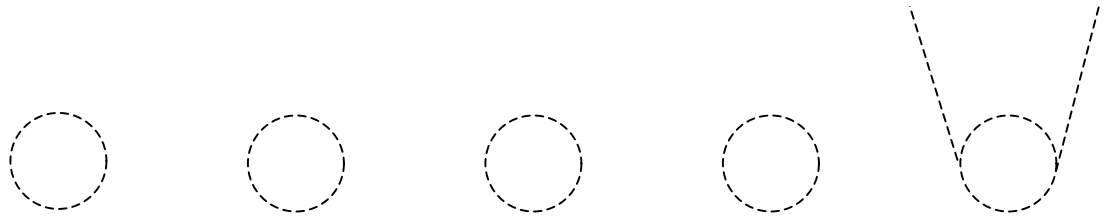
Use of this material

Department for Education
SUPPORTED BY MAYOR OF LONDON
COMPUTING AT SCHOOL EDUCATE · ENGAGE · ENCOURAGE

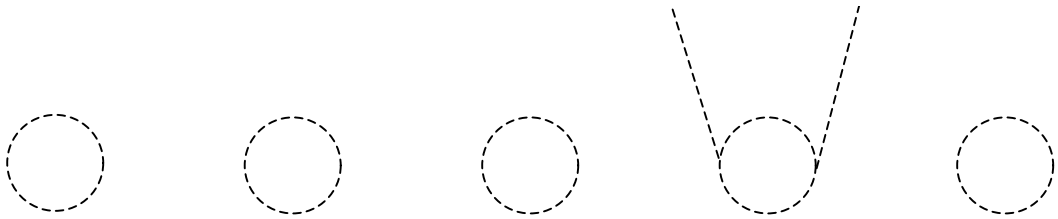
Punch cards

The following cards are ready-made punch cards that just need printing and the notches cutting out.

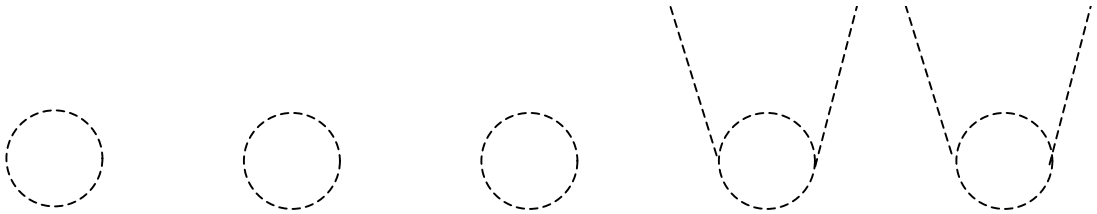
1. Print the following sheets on to thin card
    - To print without margins you will need to choose as the paper size “borderless A4” if your printer allows it.
    - If not guillotine the margins.
2. Cut out the notches / holes as indicated by the dotted lines.
3. Cut off the corner as indicated by the dotted lines
    - This is used to ensure that all the cards are the right way round before use.
4. Sprinkle talcum powder on the cards to help stop them sticking together.



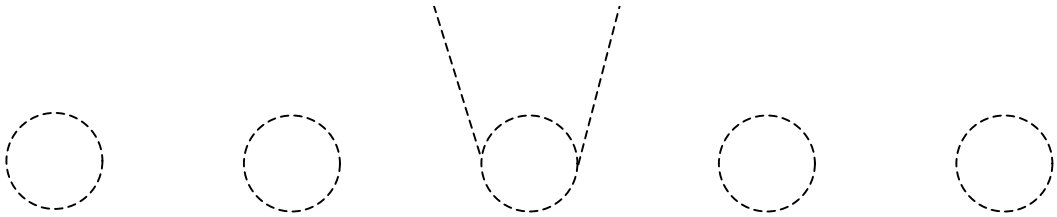
1



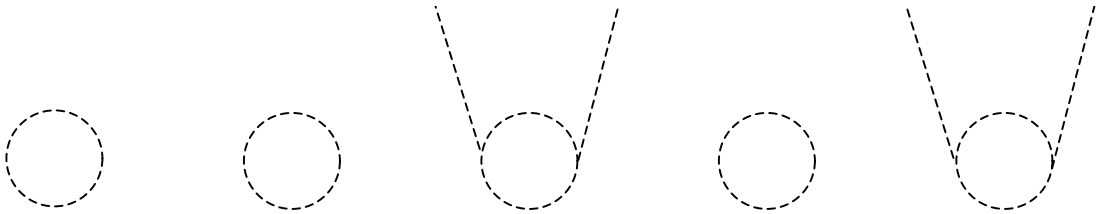
2



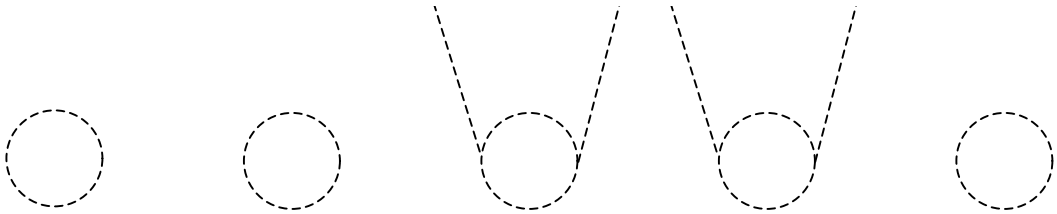
3



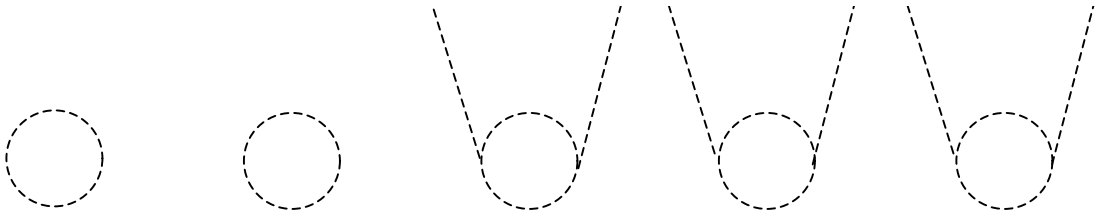
4



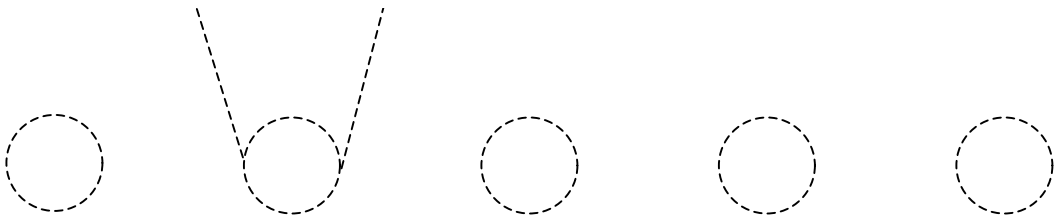
5



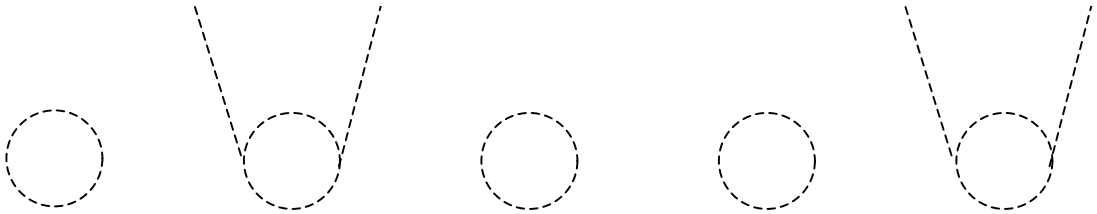
6



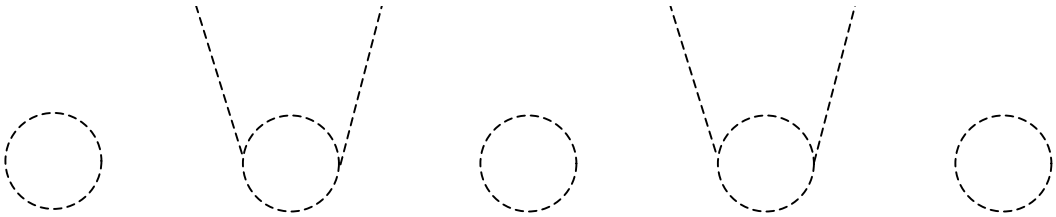
7



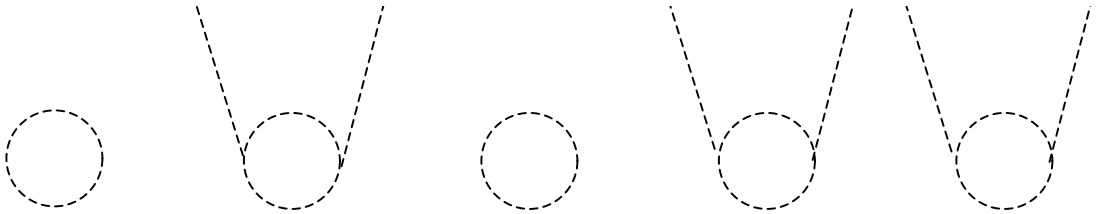
8



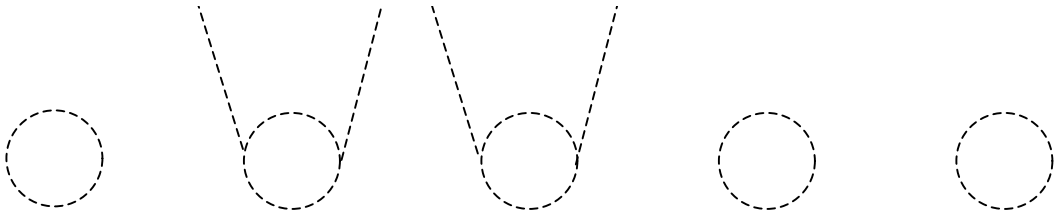
9



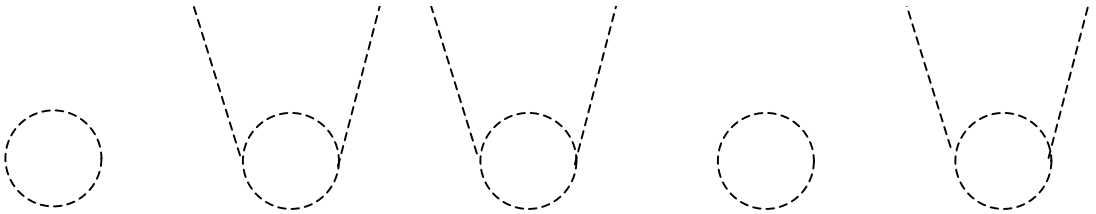
10



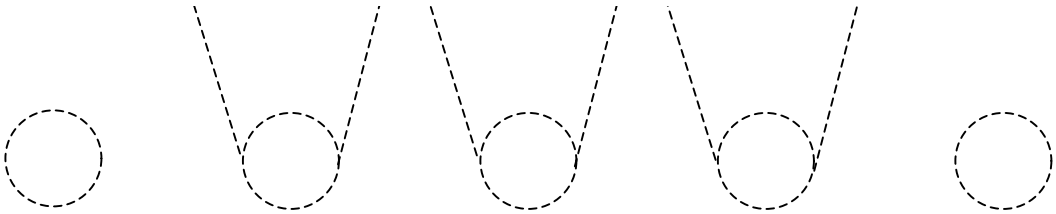
11



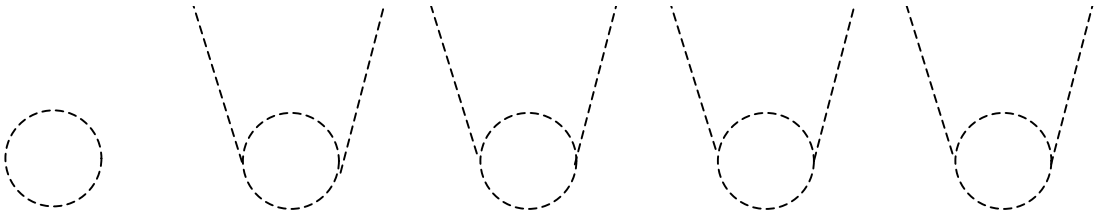
12



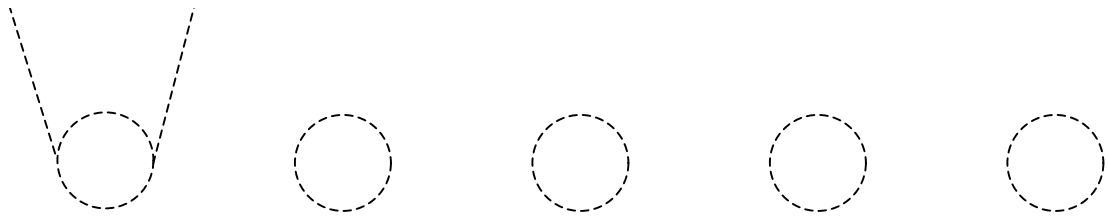
13



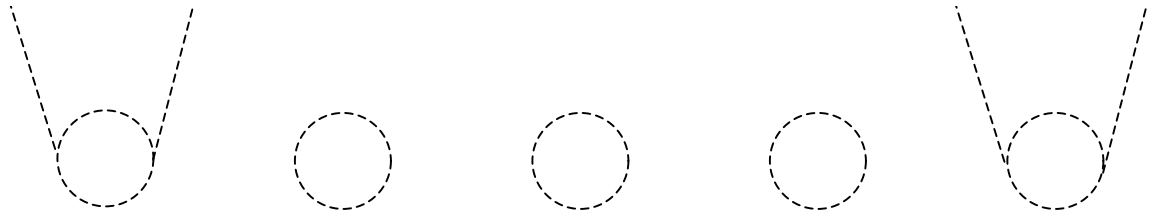
14



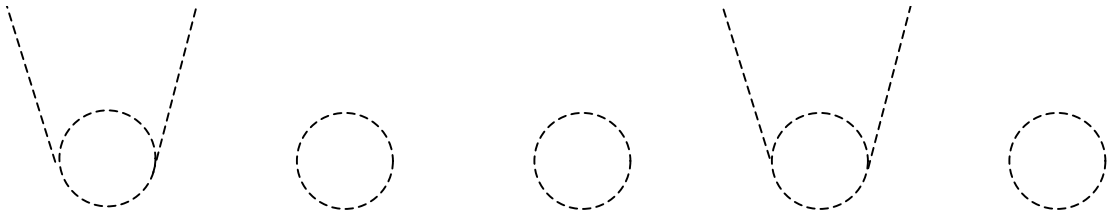
15



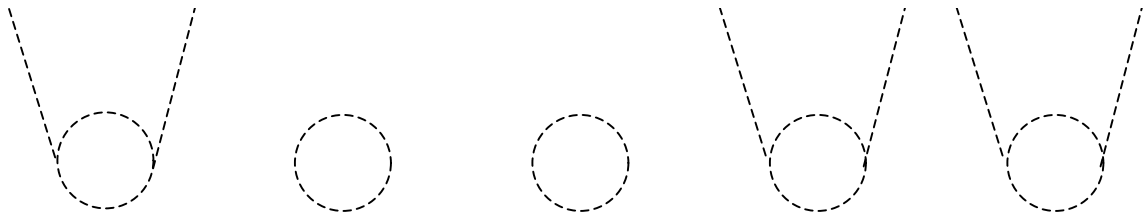
16



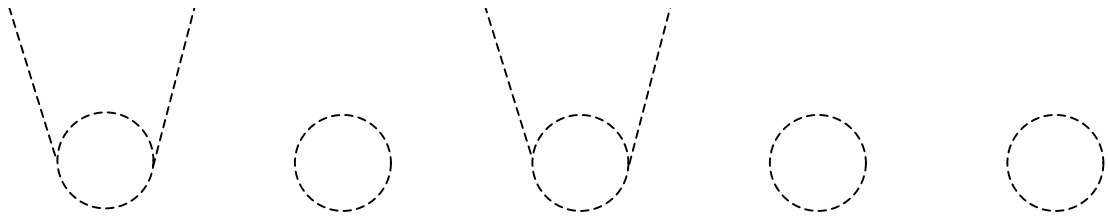
17



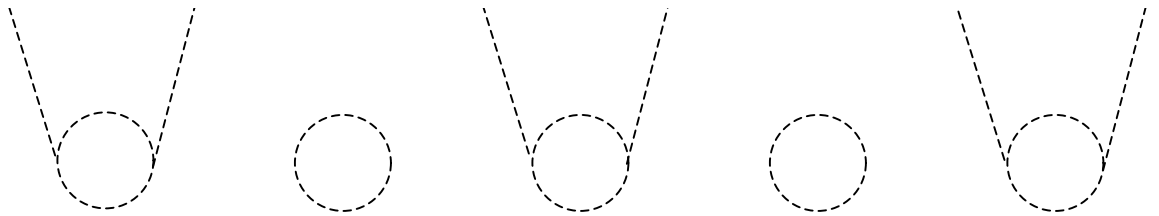
18



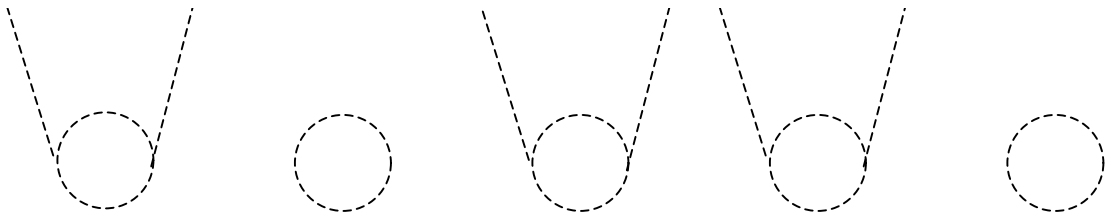
19



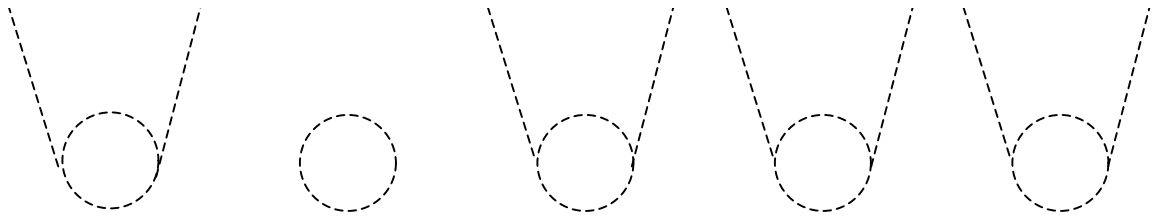
20



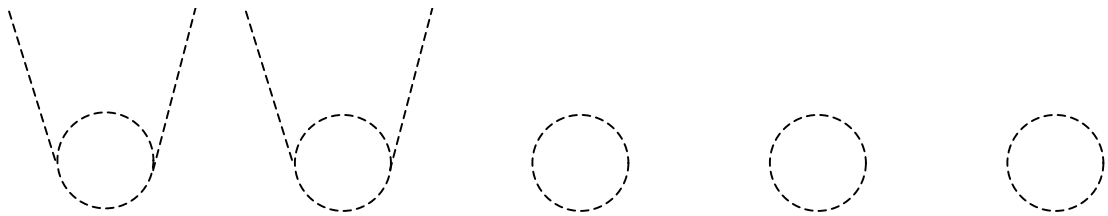
21



22



23



24