Midlands Graduate School 2007

Note Title 16/04/2007

Algorithmic Problem Solving

Roland Backhouse

(with João Ferreira, Diethard Michaelis)

Edsger W. Dijkstra (1930-2002)

"Mathematics = The Art of Effective Reasoning"

Unnecessary Naming

A farmer wants to ferry a goat, a cabbage and a wolf across a river.

His boat can carry only one of them at a time.

The wolf cannot be left with the goat.

The goat cannot be left with the cabbage.

How can the farmer get them all across?

Avoided

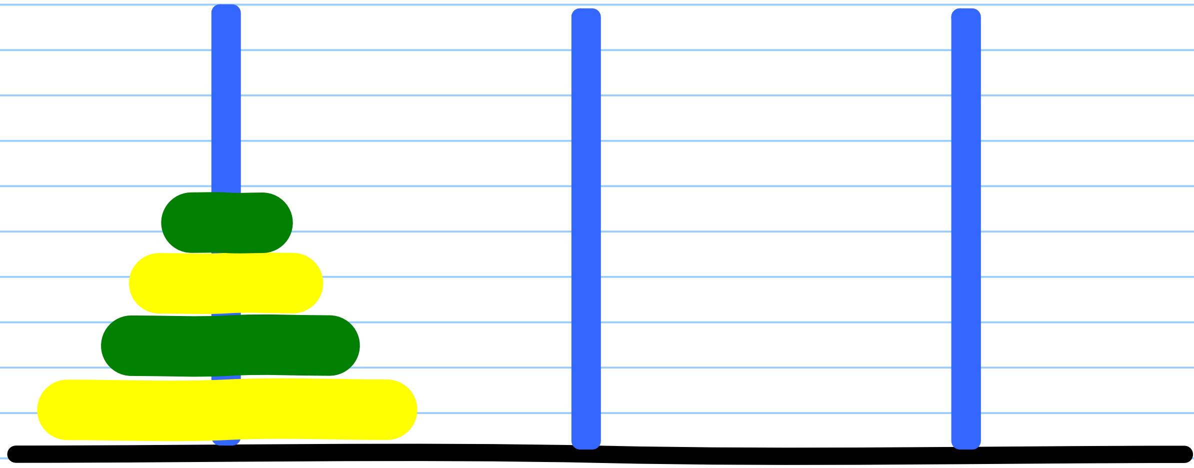
A farmer wants to ferry an alpha and two betas across a river.

His boat can carry only one of them at a time.

An alpha cannot be left with a beta.

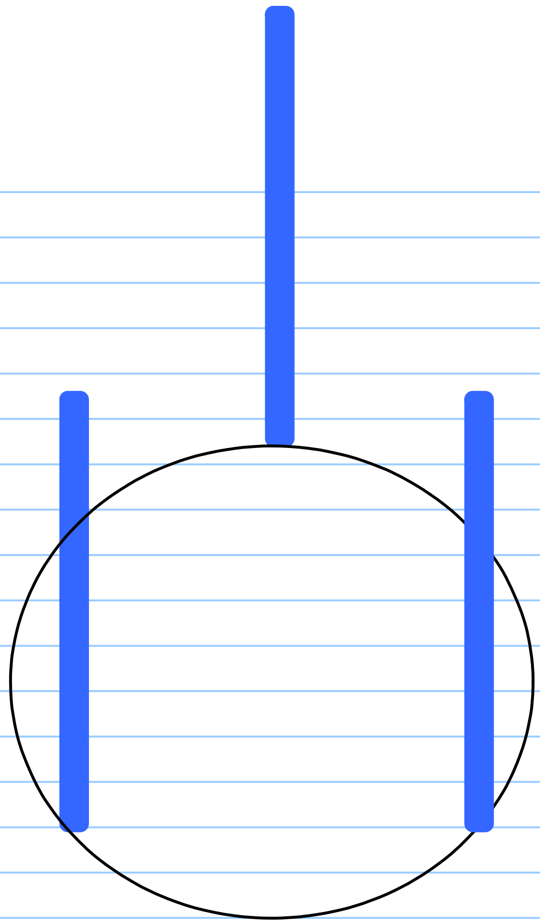
How can the farmer get them all across?

Towers of Hanoi



Move n disks from one pole to the next

- one at a time
- so that a larger disk is never on top of a smaller disk.



Arrange the poles in a circle

— avoids unnecessary naming of the poles.

Inductive Hypothesis

n	no. of disks
d	direction of movement
H	sequence of moves

$H.(n,d)$ is the sequence of moves needed to move the n smallest disks in direction d.

$$H(0,d) = []$$

$$H(n+1,d) = H(n,\neg d) \; ; \; \langle n+1, d \rangle \; ; \; H(n,\neg d)$$

Knockout Tournament

In a knockout tournament there are 1234 players. How many games are played?

Solution Introduce variables:

p no. of players

g no. of games.

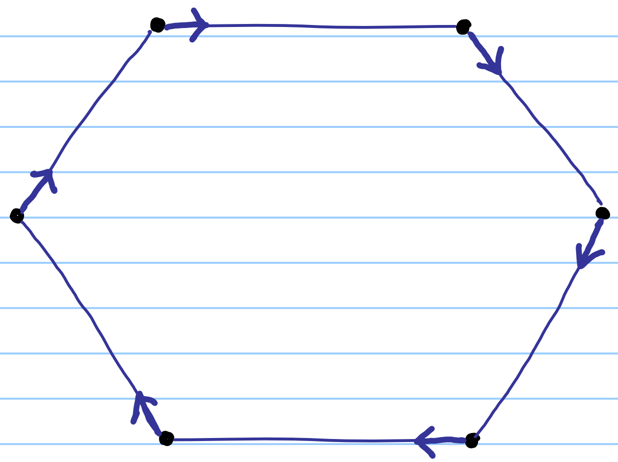
Playing a game:

Invariant:

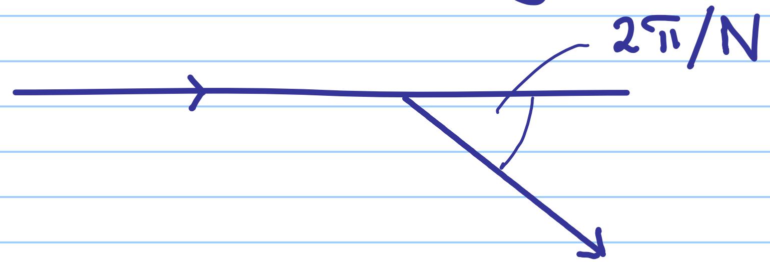
The Beetles Problem

N beetles at the corners of a regular N-gon chase each other. Each moves at constant speed v. The length of a side is s. How far does each beetle travel before they meet?

eg. $N = 6$



Consider line connecting two beetles:



Component of speed of chased beetle along line $v \times \cos \frac{2\pi}{N}$

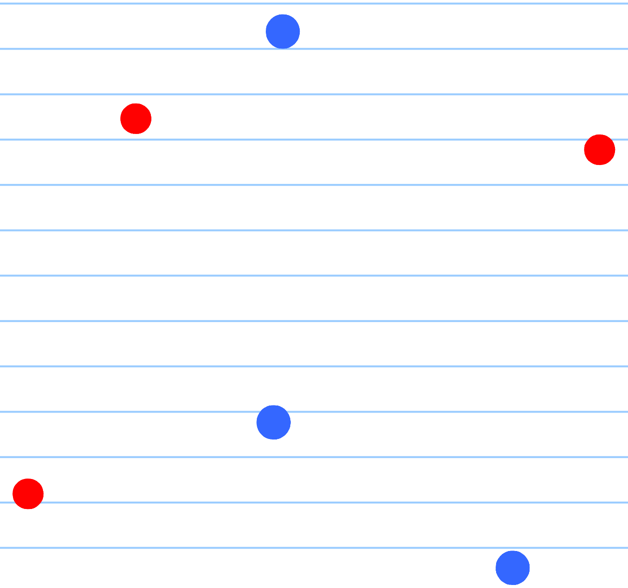
Net approach speed $v - v \times \cos \frac{2\pi}{N}$

Time taken $t = s / \left(v - v \times \cos \frac{2\pi}{N} \right)$

Distance travelled $v \times t = s / \left(1 - \cos \frac{2\pi}{N} \right)$

Making Progress

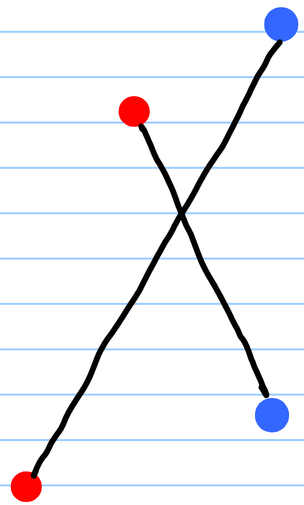
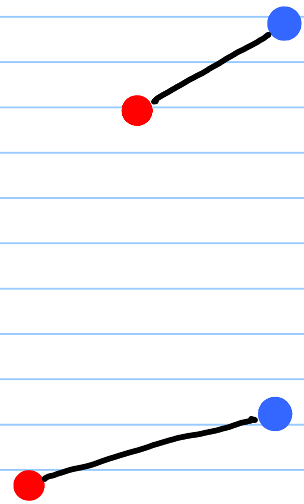
1-1 correspondence problem: given an equal number of red and blue points on a plane surface, determine a sufficient condition such that they can be put in 1-1 correspondence in such a way that the lines connecting corresponding points do not intersect.



Algorithm:

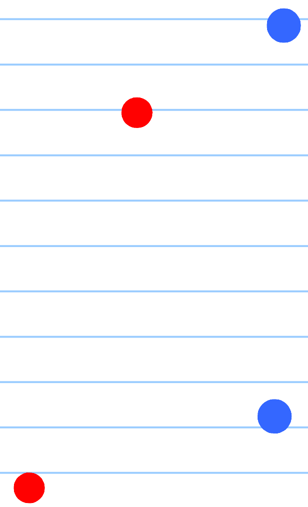
1. Construct an arbitrary 1-1 correspondence.

2. Repeatedly

   replace  by 

   until no intersections exist.

Measure of progress:



Ineffective Reasoning:

Consider the 1-1 correspondence that minimises the total length of the connecting lines.
We prove by contradiction that . . .