

ALGORITHMS

COMP3121/3821 | COMP9101/9801

COURSE INFORMATION

[Home](#)
[Course Information](#)
[Staff Contact Details](#)
[Course Textbook](#)
[Assessment](#)
[Course Schedule](#)
[Course Syllabus](#)
[Assignments, Exams and Puzzles](#)
[Course Resources](#)
[Course Evaluation and Development](#)
[Academic Honesty and Plagiarism](#)



UNSW
AUSTRALIA

University of New South Wales
School of Computer Science and Engineering

1. Puzzles related to recursion and mathematical induction

We start with a set of puzzles that are all related to **recursion and mathematical induction** we study in [Topic 1](#). have chosen interesting but also a bit tough puzzles to get you intrigued, so do not get discouraged if you find them hard.

1.1. The Party Problem

Tom and his wife Mary went to a party where four more couples were present. Not every one knew every everyone so people who did **not** know each other introduced themselves and shook hands. People that knew each other from before did not shake hands. Later that evening Tom got bored, so he walked around and asked all other guests (including his wife) how many hands they had shaken that evening, and got nine different answers. How many hands Mary shake? (Hint: you will end up doing **recursion** on the number of couples...)

1.2. The Ten Thieves Problem

Here is an "ancient" small puzzle: Two thieves have robbed a warehouse and have to split a large pile of various items without prices on them. How do they do this in a way that each thief **thinks (believes)** that he has got at least one of the value of the whole pile?

You might want to try to solve this puzzle before reading further ...

The solution is that one of the two thieves splits the pile in two parts such that he **thinks** that both parts are of equal value. The other one then chooses what he thinks is the better part. It is easy to see that both thieves have reasonable belief that they got at least a half (try to explain why).

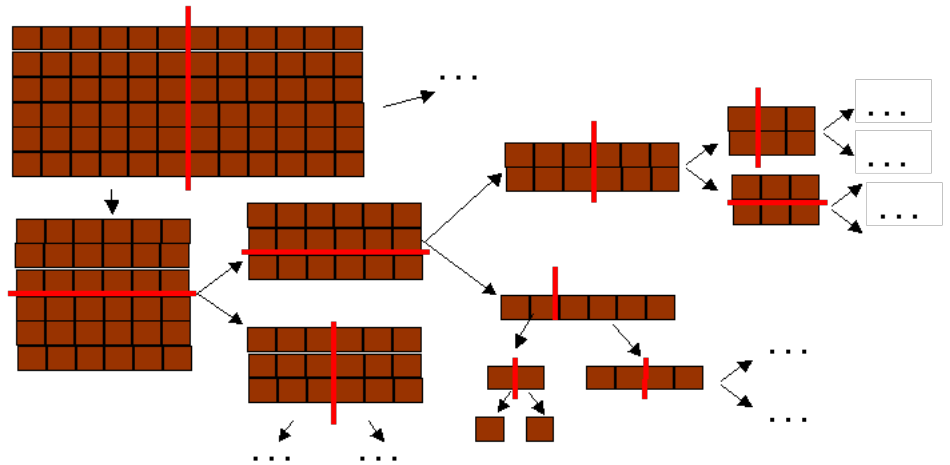
Now here is the real puzzle for you to solve: Assume that **ten** thieves have robbed the warehouse. How do they split the pile of items so that **each** thief thinks that he has got at least one tenth of the total value of the pile? (Hint: This is quite a tough one. It is an example of a **nested recursion** (a recursion within a recursion)).

1.3. Finding the False Coins

(a) We are given **27 coins** of the same denomination; we know that one of them is counterfeit and that it is lighter than the others. Find the counterfeit coin by weighing coins on a pan balance only three times. (b) We are given **12 coins** and one of them is a fake but we **do not** know if it is heavier or lighter. Can you determine which one is a fake and is lighter or heavier by weighing coins on a pan balance three times only? ((a) and (b) are perfect examples of **divide-and-conquer technique**). (c) We have **9 coins** and three of them are heavier than the remaining six. Can you find heavier coins by weighing coins on a pan balance only four times? (Hint: this is an example of the **lower bound estimation of complexity of algorithms**, i.e., of the **minimal number** of steps needed to execute an algorithm for a given input).

1.4. Breaking a chocolate

(a) Assume you are given a block of chocolate consisting of m by n squares. At each move you can break one piece into two (not necessarily equal) pieces (see the picture below). The goal is to get $m \times n$ separate squares. What is the number of moves needed to achieve this goal and how should one do it? (b) Assume now that you can put several pieces of chocolate on top of each other and break them in a single move. What is now the least number of moves needed to get $m \times n$ separate squares? (Hint: this is an example of **estimating complexity of algorithms**, i.e., the number of steps needed to execute an algorithm for a given input)



1.5. The Five Pirates

(a) There are five pirates who have to split 100 bars of gold. They all line up and proceed as follows:

- i) The first pirate in line gets to propose a way to split up the gold (for example: everyone gets 20 bars)
- ii) The pirates, including the one who proposed, vote on whether to accept the proposal. If the proposal is rejected, the pirate who made the proposal is killed.
- iii) The next pirate in line then makes his proposal, and the 4 pirates vote again. If the vote is tied (2 vs 2) then the proposing pirate is still killed. Only majority can accept a proposal. The process continues until a proposal is accepted or there is only one pirate left. Assume that every pirate :

- above all wants to live ;
- given that he will be alive he wants to get as much gold as possible;
- given maximal possible amount of gold, he wants to see any other
- pirate killed, just for fun ;
- each pirate knows his exact position in line;
- all of the pirates are excellent puzzle solvers.

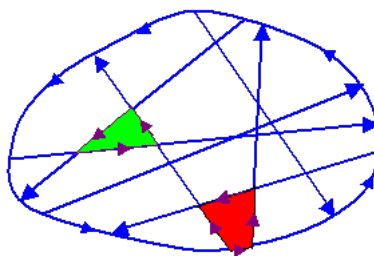
Question : What proposal should the first pirate make ?

(b) Assume now there are 10 pirates splitting 1000 pieces of gold. What should the first pirate propose ?

(An interesting puzzle - recursion seems to be the ONLY way to solve it !!!)

1.6. One Way Streets

In Elbonia all cities have a circular one-way highway around the city (in blue on the map below). All streets in the city are one-way, and they all start and end on the circular highway (see the map). A block is a part of the city that is not intersected by any street. Design an algorithm that, given a map of a city, finds a block that can be circumnavigated while respecting all one-way signs. For example, the green block has such property, but the red one does not. What is the best possible expected (i.e., average) asymptotic run time of such an algorithm? (Again a recursion, but estimating the expected run time is hard...)



[Privacy statement](#) [Copyright and Disclaimer](#) [Site Map](#) [Site Feedback](#)

© Algorithm Course, CSE, UNSW, Sydney NSW 2052, Australia. | Design by [Mohsen Rezvani](#)