Limits of Computation

17 - Common Problems Not Known to Be in $\mathbf{P}$

Bernhard Reus

Last time

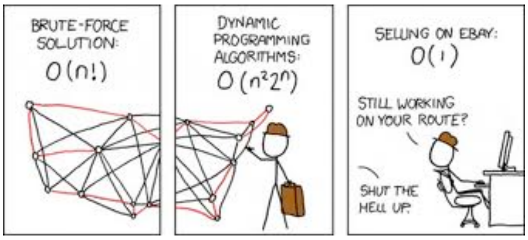
- how optimisation problems can be expressed as decision problems
- a number of famous, natural, "real-life" problems that are all provably in $\mathbf{P}$

Complexity of natural problems

THIS TIME

- We introduce some more natural (and famous) problems and discuss their runtime complexity.
- For those problems today we don't know whether they are in $\mathbf{P}$ and the question remains: *"are they feasible?"*

e.g. finding the best route



http://xkcd.com

Problems that look intractable

- we introduce more problems, all highly relevant in practice.
- which can be solved easily by *"brute-force"* search algorithms but unfortunately these algorithms do not have polynomial runtime so they do ***not*** *prove that the problems are in* $\mathbf{P}$.
- But it can be checked in $\mathbf{P}$ whether a potential solution is actually a solution (more about this next time).
- Now all problems (even optimisation problems) presented as decision problems for simplicity.

Problems of which we don't know whether they are in $\mathbf{P}$

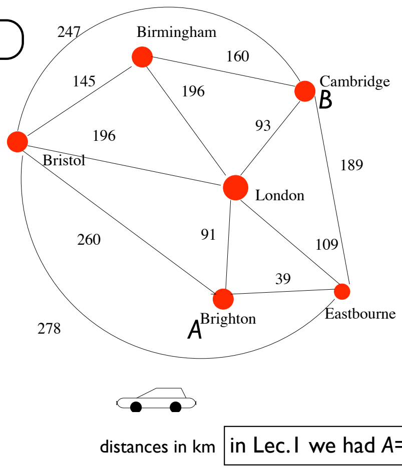
and that we conjecture may be "intractable"

Travelling Salesman Problem

see Lecture 1, now presented as decision problem

- **Instance**:
  a road map of cities, with distances attached to road segments, two cities *A* and *B* and a number *K*

- **Question**:
  Is it possible to take a trip from *A* to *B* which passes through all cities and with a total trip length less or equal *K* (km) ?



in Lec.1 we had *A*=*B*

in practice: for distribution of goods

Press coverage about TSP

guardian.co.uk

News | Sport | Comment | Culture | Business | Money | Life & style

Environment › Bees

Bees' tiny brains beat computers, study finds

Bees can solve complex mathematical problems which keep computers busy for days, research has shown

24.10.2010

Researchers found that bees could solve the 'travelling salesman's' shortest route problem, despite having a brain the size of a grass seed. Photograph: Rex Features

Bees can solve complex mathematical problems which keep computers busy for days, research has shown.

The insects learn to fly the shortest route between flowers discovered in random order, effectively solving the "travelling salesman problem", said scientists at Royal Holloway, University of London.

How many colours?

- How many colours do you need to paint a map such that no adjacent count(r)ies have the same colour?
- on the right 4 colours are used
- are 3 colours enough?
- 4 is *always* enough for maps; proved by *K.Appel & W.Haken* in 1977 ["Every planar map is 4-colourable" Illinois J. Math. 21], already observed by Englishman Francis Guthrie in 1852
  (fully machine-checked proof in 2004)

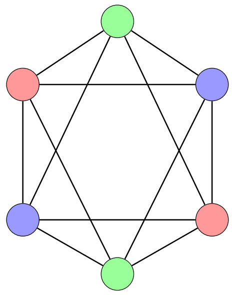
http://www.doc.ic.ac.uk/~vf05/atp/casestudy.html

maps are special graphs (planar graphs) where nodes are countries and edges between countries exist if the two countries have a common piece of borderline.

Graph Colouring Problem

- **Instance**:
  a graph and a number K of colours to be used where K is greater or equal 3 (otherwise it's easy).

for maps the only interesting (hard) case is k=3

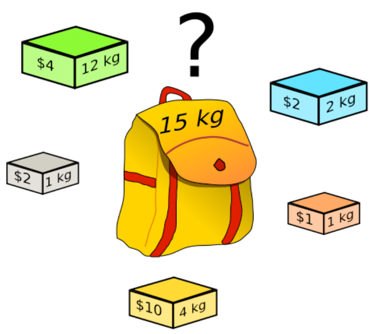


- **Question**:
  Can one colour the graph in a way that each pair of adjacent nodes (connected by an edge) uses two different colours (using only the K colours provided)?

in practice: needed for register allocation in compilers, for traffic channel assignment (networks and air); conflicting registers/planes/jobs/frequencies have a common edge, colours = how many resources needed at any time?

0-1 Knapsack Problem

- **Instance**:
  a set of n items with sizes $\{w_1,\ldots,w_n\}$ and *profits* $\{p_1,\ldots,p_n\}$, the size of a knapsack W, and a threshold K. All values are positive natural numbers.



javaingrab.blogspot.co.uk/2014/07/implementation-of-01-knapsack-problem.html

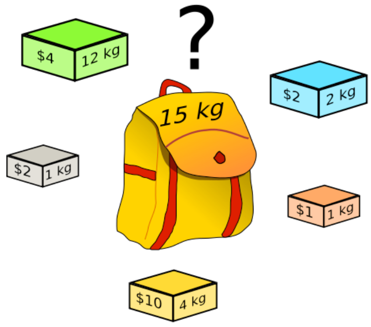
- **Question**:
  Is there a way a subset of the n items can be packed such that their profits are at least K without exceeding the capacity W?

in practice: truck loading, capacity planning

0-1 Knapsack Problem (cont'd)

Can be also represented via inequalities:

$$\sum_{i=1}^{n} w_i \times x_i \leq W \qquad \sum_{i=1}^{n} p_i \times x_i \geq K$$



where the x_i prescribe the solution and must be all 0 or 1 (hence the name)

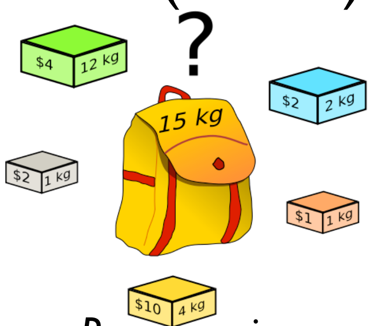
Proposition 17.1. *Any instance of the 0-1 Knapsack Problem can be solved in* $\mathcal{O}(n^2 \times p_{\max})$, *where (as in Def. 17.4) n is the number of items and $p_{\max}$ is the largest profit value.*

Why is Knapsack not in $\mathbf{P}$ then?

0-1 Knapsack Problem (cont'd)

Can be also represented via inequalities:

$$\sum_{i=1}^{n} w_i \times x_i \leq W \qquad \sum_{i=1}^{n} p_i \times x_i \geq K$$



where the x_i prescribe the solution and must be all 0 or 1 (hence the name)

0-1 Knapsack is thus an instance of *Linear Programming* which we know is in $\mathbf{P}$.

But all solutions must be *integers*, so we conclude that it is also unknown whether *Integer Linear Programming* is in $\mathbf{P}$

Complexity of problems?

- We don't know whether the problems on previous slides are in $\mathbf{P}$.

n = size of graph (number of nodes i.e. cities)

- Simple *"generate and test"* algorithms are exponential. For instance check all TSP paths: there are about *(n-1)!/2* so time bound even worse than exponential *2^N.*

Stirling's formula: $n! \sim \sqrt{2\pi n}(\frac{n}{e})^n$

- *We* define a new class for which we know that it contains all the problems just presented: **NEXT TIME!**

END

Next time:
Programs than can "guess" and a *One Million Dollar* Question!