What type of problem do we have to solve?

1. Give three different notions relating to general problem solving that are key concerns in Computer Science? [3]

Solution:

- Computation
- Resource
- Correctness

2. What is abstraction in relation to real-world problem solving? How does abstraction in Computer Science differ from abstraction in other subjects? [4]

Solution:

- Hiding information, such as to make the problem easier to solve
- In Computer Science, abstraction is concerned with information hiding, whereas in other subjects abstraction is more concerned with information neglect

3. What is the real-world travelling salesman problem? Explain how this problem is usually abstracted so that it is amenable to computational solution. Illustrate how this abstraction might not be in keeping with reality. [5]

Solution:

- Computing the shortest tour of a given collection of cities so as to same a resource of interest
- If just shortest distance is required, then it could be abstracted to the cities and the distances between them
- If time was important, then distributions that model traffic conditions rather than distances would need to be created for each pair of cities
- This might not be in keeping with reality because the distribution for traffic is a complicated one and may not be modelled correctly.

4. What is the real-world map-colouring problem? Explain how we might abstract this problem as a graph colouring problem. What is a planar graph? Explain how planar graphs and maps are related. Is every graph a planar graph? [7]

Solution:

- Given a plane map of regions, each contained within a continuous border, can you colour the regions of the map with 3 colours so that if any two regions touch they must be coloured differently?
- This can be abstracted into a graph, as all the important information is just if two regions touch
- **Planar Graph** - A graph that can be drawn in a plane without any graph edges crossing
- Every map can be turned into a planar graph
- Not every graph is a planar graph, some will have edges that cross

5. What is the real-world scheduling problem? Explain how we might abstract this problem as a graph colouring problem. [4]

Solution:

- You have a set of jobs that need to be allocated to t (equal length) time slots. Jobs can be scheduled in any time slot, but some pairs of jobs are in conflict as they cannot be scheduled in the same time slot. Can we decide whether there is a schedule for the jobs so that the number of time slots is at most t?
- Jobs are abstracted as vertices, and two vertices are joined by an edge if the corresponding jobs are in conflict
- There is a schedule of the jobs in t slots so that no conflicting jobs occupy the same slot if, and only if, the corresponding graph can be coloured using t colours

6. What is the real-world register allocation problem? Explain how we might abstract this problem as a graph colouring problem. [4]

Solution:

- A large number of program variables are allocated to a small number (k) of CPU registers
- Not all program variables are in use at the same time
- If two program variables are in use at the same time, then they have to be assigned to different registers
- Can we decide whether every program variable can be assigned to one of our k registers so that if two program variables are in use at the same time then they are necessarily assigned to different registers
- Program variables are abstracted as vertices, two vertices are joined by an edge if the corresponding program variables are in use at the same time
- There is an allocation of program variables to k registers so that if any two program variables are in use at the same time then they have to be allocated different registers if, and only if, the corresponding graph can be coloured using k colours.

7. What is the real-world frequency assignment problem? Explain how we might abstract this problem as a graph colouring problem. [4]

Solution:

- Consider a network of radio transmitters, each of which must be assigned one of f operating frequencies
- If two nearby transmitters are operating on the same frequency then they have the potential to interfere with each other, the frequencies assigned to such pairs of transmitters are required to be distinct
- Can we decide whether we can allocate a frequency, from a set of f frequencies, to every transmitter, so that any pair of nearby transmitters mist have different frequencies
- Transmitters are abstracted as vertices, and two vertices are joined by an edge if the corresponding transmitters are nearby each other.
- There is an assignment of f frequencies to transmitters if, and only if, the corresponding graph can be coloured using f colours

8. What is the essential rule to bear in mind when developing an abstraction of a real-world problem? [2]

Solution:

The abstractions mirror reality sufficiently closely so that any conclusions reached remain valid as regards the real world problem

9. Give a precise definition of a decision problem. [2]

Solution:

A decision problem D consists of:

- a set of instances I
- A subset $Y \subseteq I$ of yes instances

10. What is a graph? What is a proper colouring of a graph and an optimal colouring of a graph? Describe two different algorithms (in English) that enable you to colour graphs. Explain whether or not your algorithms yield an optimal colouring (you may use graphs to illustrate your answer if you wish). Show how your algorithms proceed when you wish to decide whether the graph in Fig. 1 can be coloured using 4 colours [20]

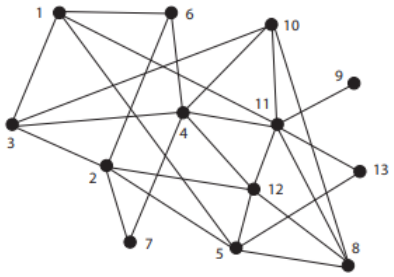


Fig. 1: a graph.

Solution:

A graph, G=(V,E), has a finite set V of vertices and a set E of pairs of distinct vertices where no edge is repeated and (in an undirected graph) (u,v)=(v,u). A proper colouring is a colouring in which any two adjacent vertices are coloured differently, the optimal colouring is a proper colouring that uses the smallest number of colours.

Algorithm one: work through the nodes in increasing order and colour using the smallest available colour. This will always yield a proper colouring, but may not yield an optimal colouring, however if it does yield an optimal colouring, that colouring is correct.

Algorithm two: chose a node that can be coloured using the lowest colour possible, if there are multiple nodes that can be coloured with that colour, then chose the node with the lowest vertex order. Once no other nodes can be coloured using that colour, move onto the next colour.

Both algorithms produce an identical 3-colouring for the graph.

1:1, 2:1,3:2,4:1,5:2,6:2,7:2,8:1,9:1,10:3,11:2,12:3,13:1

11. What is the famous Four Colour Theorem in graph theory? [2]

Solution:
Every planar graph can be properly coloured using just 4 colours

12. Give a (not necessarily precise) definition of what it means for an algorithm to be greedy. [2]

Solution:
An algorithm that works through a list "greedily" choosing the best thing to do

13. What is a crown graph on 2n vertices? What is the smallest number of colours that can be used so as to obtain a proper colouring of the crown graph? [2]

Solution:
In a crown graph with 2n vertices, there is a vertex ordering so that our greedy colouring algorithm uses n colours when 2 is optimal

14. Sometimes we can decide whether a graph can be coloured with a certain number of colours, k say, by transforming the graph into a smaller graph so that the original graph can be k-coloured if, and only if, the transformed graph can. On occasion, we can iteratively do this so that we get a graph so small that we can trivially see whether or not it can be k-coloured. Give an example of such a transformation that we might use in this way. Apply your transformation to decide whether the graph in Fig. 1 can be coloured using 4 colours. [8]

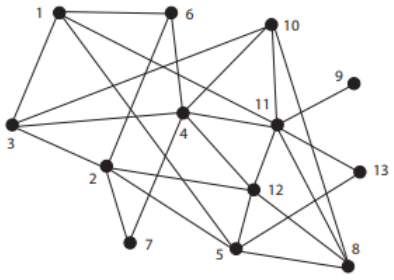


Fig. 1: a graph.

Solution:
If a vertex has less than n neighbours (where n is the number of colours to be tested), then it can be removed from the graph, and if the remaining subgraph can be coloured with n colours then the larger graph can also be coloured using n colours.

By repeatedly performing this on figure 1 you get to the point where all the nodes have been removed, this can trivially be 4 coloured

15. What do we mean by a brute-force algorithm to decide whether a graph can be coloured with 4 colours? [2]

Solution:
An algorithm that will try all possible 4 colourings and see if it is legal or not

16. Suppose that we are given two lists of symbols (possibly of different lengths). Explain how we order these two lists according to the lexicographic ordering [3]

Solution:

- We compare the two lists by working down the lists entry bu entry until two corresponding entries differ
- The list with the least entry is determined to be the least of the two lists
- If we never find two different corresponding entries but run out of one of the lists then this shortest list is the least list

17. Give a precise definition of a search problem and of a solution of a search problem. [5]

Solution:
A search problem S consists of:

- A set of instances I
- A set of solutions J
- A binary search relation $R \subseteq I \times J$

In order to solve a search problem, for any instance $x \in I$ we need to:

- find a solution $y \in J$ such that $(x, y) \in R$ if there is one, or
- Answer no if there is no solution y for which $(x, y) \in R$

18. Explain how there is always a decision problem corresponding to any search problem. [1]

Solution:

If you can find a solution to the search problem, the result is yes to the decision problem

19. Give a precise definition of an optimisation problem and of a solution of an instance of an optimisation problem. [5]

Solution:

An optimisation problem O is defined as follows:

- It consists of a set of instances I
- for every instance $x \in I$ there is a set of feasible solutions f(x)
- For every instance $x \in I$ and for every feasible solution $y \in f(x)$ there is a value $v(x, y) \in \mathbb{N} = \{0, 1, 2, ...\}$ giving the measure of the feasible solution y for the instance x
- There is a goal which is either min or max

To solve an optimisation problem, given $x \in I$, we need to find a feasible solution of maximum or minimum measure. It may be the case that:

- There are no feasible solutions for an instance
- For a maximisation problem, there are feasible solutions of increasingly large measure

20. What is the Graph 3-Colouring (decision) problem and what is the Graph Colouring (optimisation) problem? [4]

Solution:

- You are given a plane map of regions, each contained within a continuous border
- For any pair of regions, either
  - They share no part of a border
  - Exactly one point of a border, when they do not touch
  - They share a segment of some border, when they do touch
- Can you colour the regions of the map with 3 colours so that if any two regions touch they must be coloured differently

21. What is an independent set in a graph? What is the Independent Set (optimisation) problem? [3]

Solution:

- An independent set is a set of vertices in a graph, no two of which are adjacent
- The independent set problem:
  - An independent set U in a graph $G = (V, E)$ is a subset $I \subseteq V$ of verticies so that no vertex in U is joined by an edge to any other vertex of U
  - A maximum independent set in G is an independent set of the greatest size possible

22. What is the difference between a maximum independent set and a maximal independent set in a graph? [2]

Solution:

A maximal independent set that is not contained in any other independent set

23. Outline a real-world application of finding independent sets in graphs (you should explain clearly how finding independent sets is related to solving the real-world problem). [5]

Solution:

- The sports day problem is the problem of allocating events to slots such that people participating in multiple events will be able to compete in all events.
- An independent set would have all athletes able to take part in all competitions

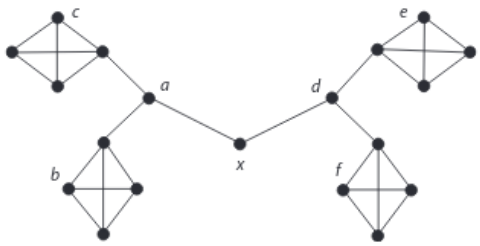
24. Outline a greedy algorithm for finding an independent set in a graph. Does your algorithm solve the Independent Set (optimisation) problem (optimally)? (If so then you should explain why; if not then you should provide a counter-example.) Show how your algorithm proceeds on the graph in Fig. 1. [10]

Solution:

- We chose the vertex v of our graph with the fewest number of neighbours and add it to the independent set I
- We remove v (and its incident edges) as well as all neighbours of v (and their incident edges) from our graph and repeat (with our new, smaller graph)

25. Give a graph where as to how you implement your greedy algorithm, above, matters; that is, where when different possible choices for putting some vertex in your independent set yields independent sets of different size. [8]

Solution:



26. Give two illustrations of principles of Computational Thinking in the context of problem solving. [2]

Solution:

- Algorithmic graph theory allows you to analyse graphs, which are used in many of the sciences
- The Travelling Salesman Problem can be used for shipping companies to determine optimum distribution routes

27. Outline a greedy algorithm for finding a tour in a given collection of cities. Does your algorithm solve the Travelling Salesman (optimisation) problem (optimally)? (If so then you should explain why; if not then you should provide a counter-example.) Show how your algorithm proceeds on the following instance of the Travelling Salesman problem (the city names are in bold and the distances between 2 cities are given in the table): [10]

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	4	11	23	3	6	2	2
1	4	0	4	4	4	5	4	4
2	2	4	0	12	1	13	3	3
3	32	4	11	0	2	3	4	5
4	10	8	1	12	0	7	6	9
5	2	2	2	22	11	0	6	7
6	4	5	9	5	6	5	0	8
7	3	5	4	6	9	10	7	0

Solution:

- Start at some node
- Iteratively move to the nearest node
- Output the tour obtained

One example is made using the paths shown below

	0	1	2	3	4	5	6	7
0	⓪	4	11	23	3	6	②	2
1	4	0	④	4	4	5	4	4
2	2	4	0	12	①	13	3	3
3	③②	4	11	0	2	3	4	5
4	10	8	1	12	0	⑦	6	9
5	2	2	2	22	11	0	6	⑦
6	4	⑤	9	5	6	5	0	8
7	3	5	4	⑥	9	10	7	0

This gives a tour of 0,6,1,2,4,5,7,3,0 which has length of 58, and follows the above algorithm However in a few places there are multiple choices of where to go, as they all have the same length, so if the below choices are made:

	0	1	2	3	4	5	6	7
0	0	4	11	23	3	6	②	2
1	4	0	4	4	4	⑤	4	4
2	2	4	0	12	1	13	3	③
3	32	4	11	0	②	3	4	5
4	10	8	①	12	0	7	6	9
5	②	2	2	22	11	0	6	7
6	4	5	9	⑤	6	5	0	8
7	3	⑤	4	6	9	10	7	0

This is following the algorithm as well, but produces the tour of 0,6,3,4,2,7,1,5 which has length 25, so the first time was not optimal. This tour is not necessarily optimal, but given it is smaller than the first attempt then the first attempt cannot be optimal.