The Equal Groups Problem

Background:

Many real world problems can be generalised into what is often called ‘grouping’ problems in academia. The idea in such problems is to divide a pile of ‘items’ into groups so that the groups are equal in some way :

1. Allocating work amongst staff teams so that workload is equal
2. Grouping containers on ships so that the weight on each ship is equal
3. Grouping deliveries onto vehicles so that the volume in each vehicle
4. Distributing computer jobs across a set of machines such that the load on each machine is equal

Such problems share the same characteristics of some of the tasks discussed in the lecture – there are many, many ways of assigning items to groups and it is likely to be impossible to evaluate each possible solution individually.

Task 1

For each of the cases above – determine the item that is to be divided between groups, what a group represents, and write down a metric by which you could determine whether or not the items had been successfully divided. Fill in your answers in the table below:

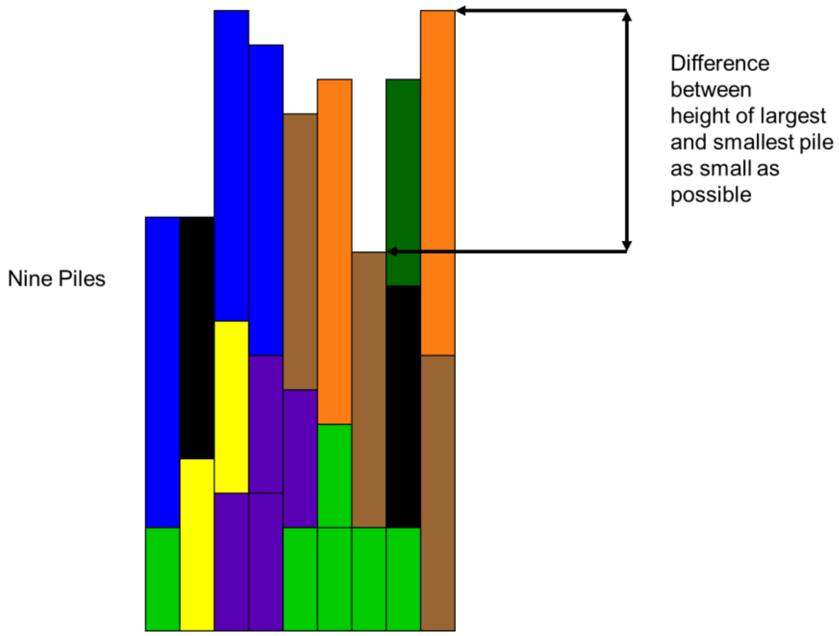
Problem	Item	Group	Metric
1			
2			
3			
4			

Task 2

You are now going to experiment with solving some problems by yourself!

You will be given a set of coloured 'blocks'. Imagine each block represents an item of work, with the length of the block proportional to the time the work will take to complete. You have to distribute the items to 9 groups or 'teams'. The goal is to find a solution in which every time has exactly the same amount of work. Note:

- Every item must be distributed to a group – you can't leave any items out!
- There must be exactly 9 groups
- If it is impossible to find a perfect solution, then you should find an assignment that minimises the difference in work between the group with the largest amount and the group with the smallest amount.



Problem 1

For the first problem, you are given these blocks (check you have exactly this)

- 2 orange
- 3 blue
- 3 brown
- 2 black
- 1 dark green
- 2 yellow
- 4 purple
- 6 light green

When you have found the best solution, write down a set of rules (heuristics) that you used to solve the problem. The heuristics should be generic in that they could be applied to other problems. For example:

Suitable Heuristic: Sort the items into decreasing size
Suitable Heuristic: Place the smallest item in a group first
….

We are not looking for heuristics such as 'put the blue item in a group and add the yellow one' – because this rule is only applicable to this particular problem and could not be used to solve a new problem instance.

Write down your rule(s) here:

Problem 2
Using the rules you have just described, try and solve this problem. Again you should sort the items into 9 groups. *Make sure you have exactly this combination of blocks before you start.*

- 2 orange
- 3 blue
- 3 brown
- 1 black
- 3 dark green
- 3 yellow
- 4 purple
- 4 light green

Now answer these questions:

1. What is the best solution you can find ?
2. Did your previous rules work ?
3. If necessary, modify your rules and write down the new rule.

Problem 3

Finally, take your (possibly modified) set of rules and try and solve this problem – again make sure you have the correct set of blocks before you start.

- 4 orange
- 3 blue
- 3 brown
- 3 black
- 3 dark green
- 1 yellow

(note no light-green or purple blocks)

If necessary, modify your rules again.

Task 3

Write down a generic algorithm that could be used to solve any instance of this problem. ***Try and write your algorithm in the form of pseudo-code.*** Test it by repeating the examples above or inventing your own. Pass the algorithm to another class member and see if they can follow it !

The algorithm should be applicable to any set of blocks with any number of groups, and should not be specific to the particular problem you were using this morning.

This is an important concept for the module – we will be looking at generic problem-solvers – the solver should be able to take any instance of a problem and output a solution to that problem.

Task 4

In the lecture, we discussed how it was important to be able to represent a problem in a way that it was easy to write down a list of potential solutions (regardless of quality). For example, we might represent a solution as a string of integers, binary values or real-numbers. For example ….

- A solution to the knapsack problem with n items can represented as a bit string of length 10. This lets us write down any number of solutions as a string of length n where each value is assigned to be 0 or 1
- A solution to the travelling salesman problem with n cities can be written down as a permutation of n integers

Consider a grouping problem in which there are 10 items that have to be assigned to 4 possible groups. Think about how you might represent a solution to this problem in a string format, and then using your representation, write down 5 possible solutions. Remember that we are not making any prediction about the quality of these solutions – they are merely potential solutions to the problem.

Some hints:

a) Consider labelling each block with a unique tag (there were are 10 blocks) – does this help you come up with a representation ?
b) You need to represent the solution as string – how long will it be ?
c) There are several ways you can consider writing down a representation:
   i) think about the problem from the perspective of each block – how do I write down where each block goes in any solution ?
   ii) think about the problem from the perspective of the 4 groups – how do I know which blocks each group contains in any solution ?

Bring your solutions to class next week as we will be discussing them in the 2nd part of the next lecture.