Problem 3: Clothing coordination (25 points)

This is a very difficult problem that will require some thought. You will be working through a

problem that recommender sites deal with. Imagine an online store that sells clothing. The

website has a list of items that pair well together. Let’s say shirt number 1 is usually bought with

pants number 6 or shoes number 10. If there is only 1 item in the shopping cart, it is easy to

recommend those pants or those shoes and many websites do this. However, let’s say a user

wants to buy shirt number 1 and shirt number 2. What are the items that pair well with *both* 1

and 2? This is the motivation behind this problem: a recommender system based on multiple

input values.

The input will be constructed in the following fashion with the following list for 1 particular item

*appearing on one line* ( *disregard the line break in this document* ):

ClothingID : ClothingIDOne ClothingIDTwo ClothingIDThree… !

NotClothingOneID NotClothingTwoID NotClothingThreeID…

Each item list will be on its own line (see example below). All of the items that pair well with

ClothingID are listed before the exclamation point. All of the items that don’t pair well with

ClothingID are listed after the exclamation point. All IDs will be integer values to keep things

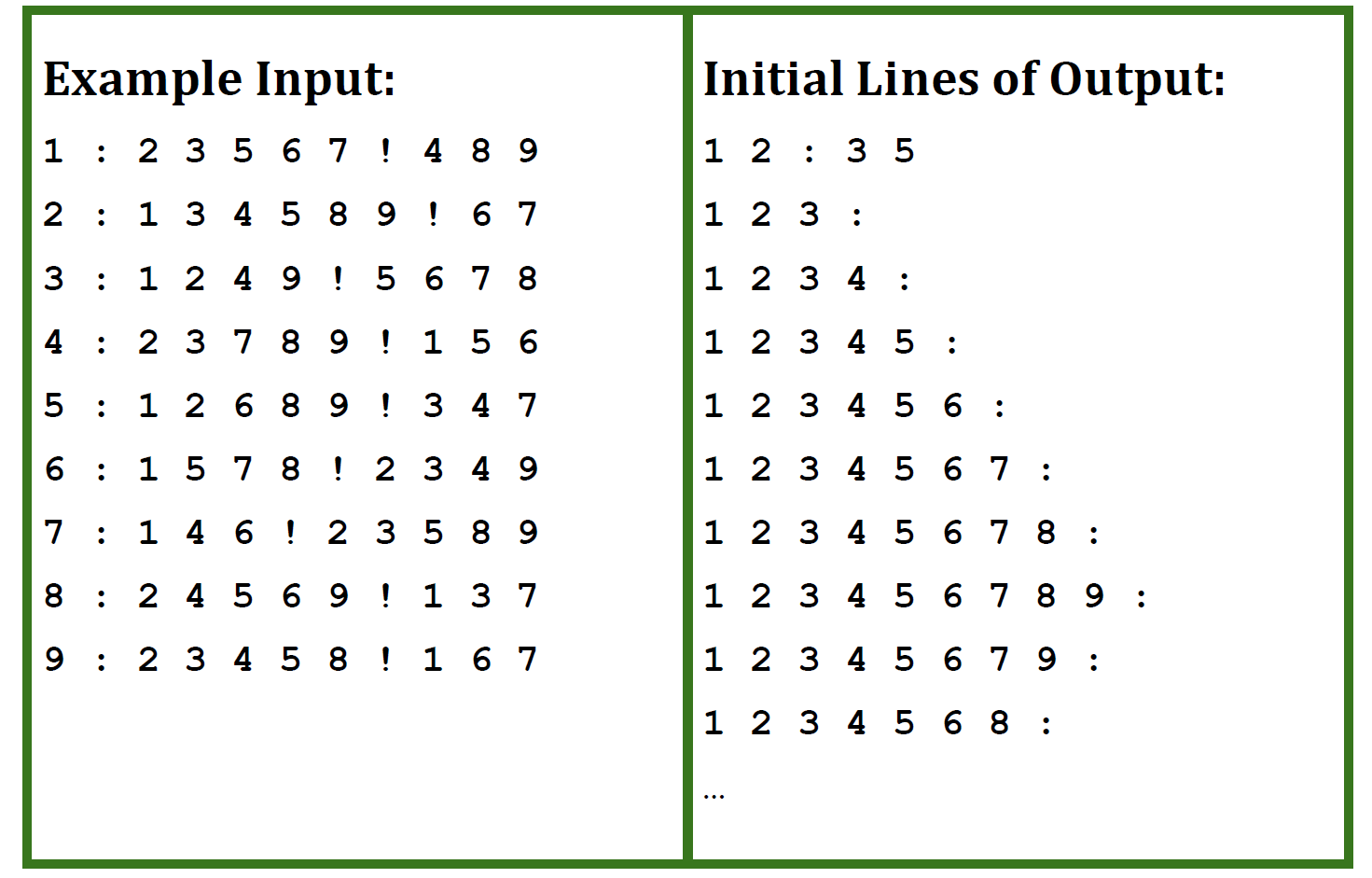
simple. If clothing item 1 pairs well with clothing item 2, then clothing item 2 also pairs well with

clothing item 1. The goal is this: produce all combinations of clothing items and the clothing

items that they all pair well with. The following pages walk you through examples and output.

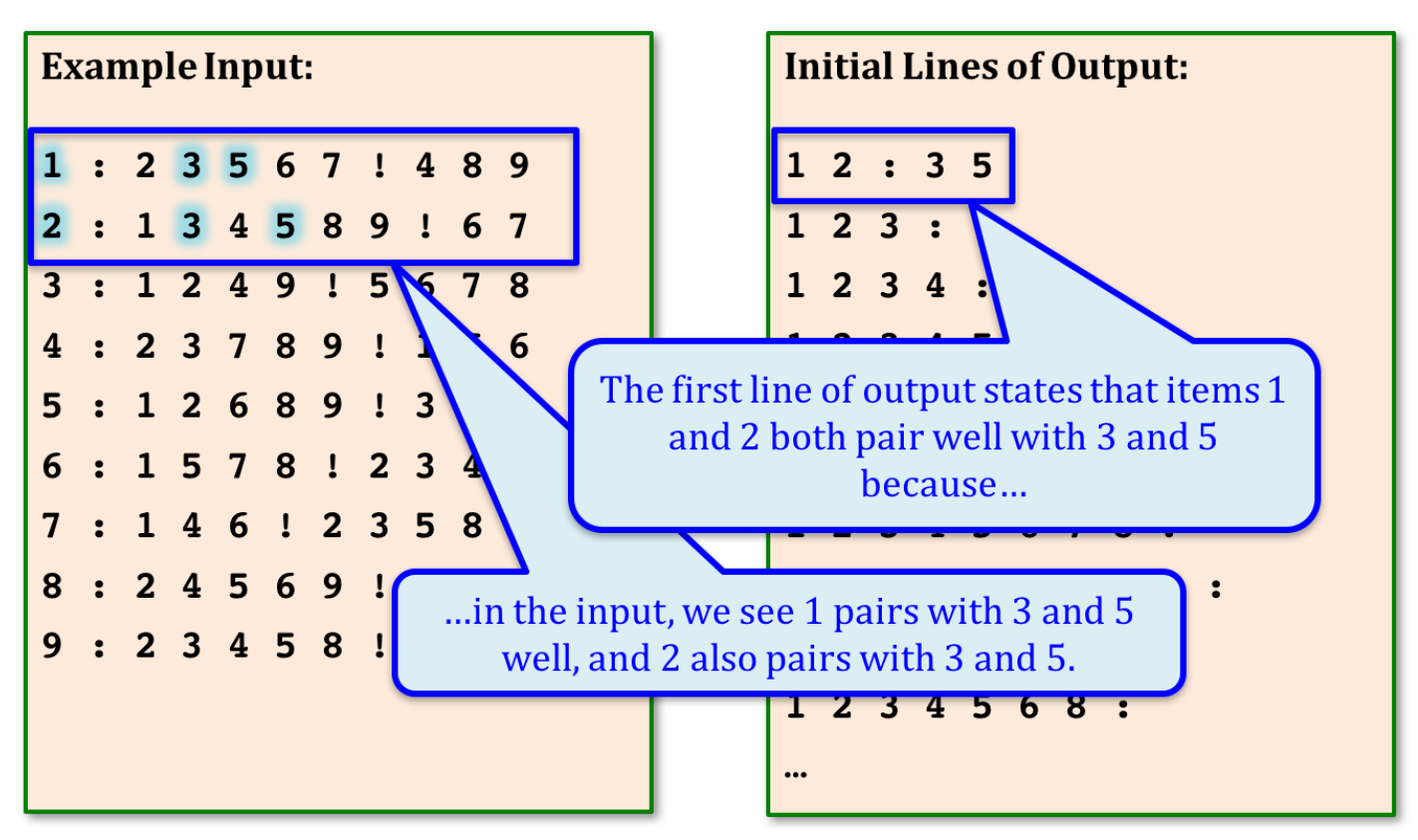
As a starting point, consider the example input and starting output shown below. The next few

pages go into more detail about how to interpret the output.

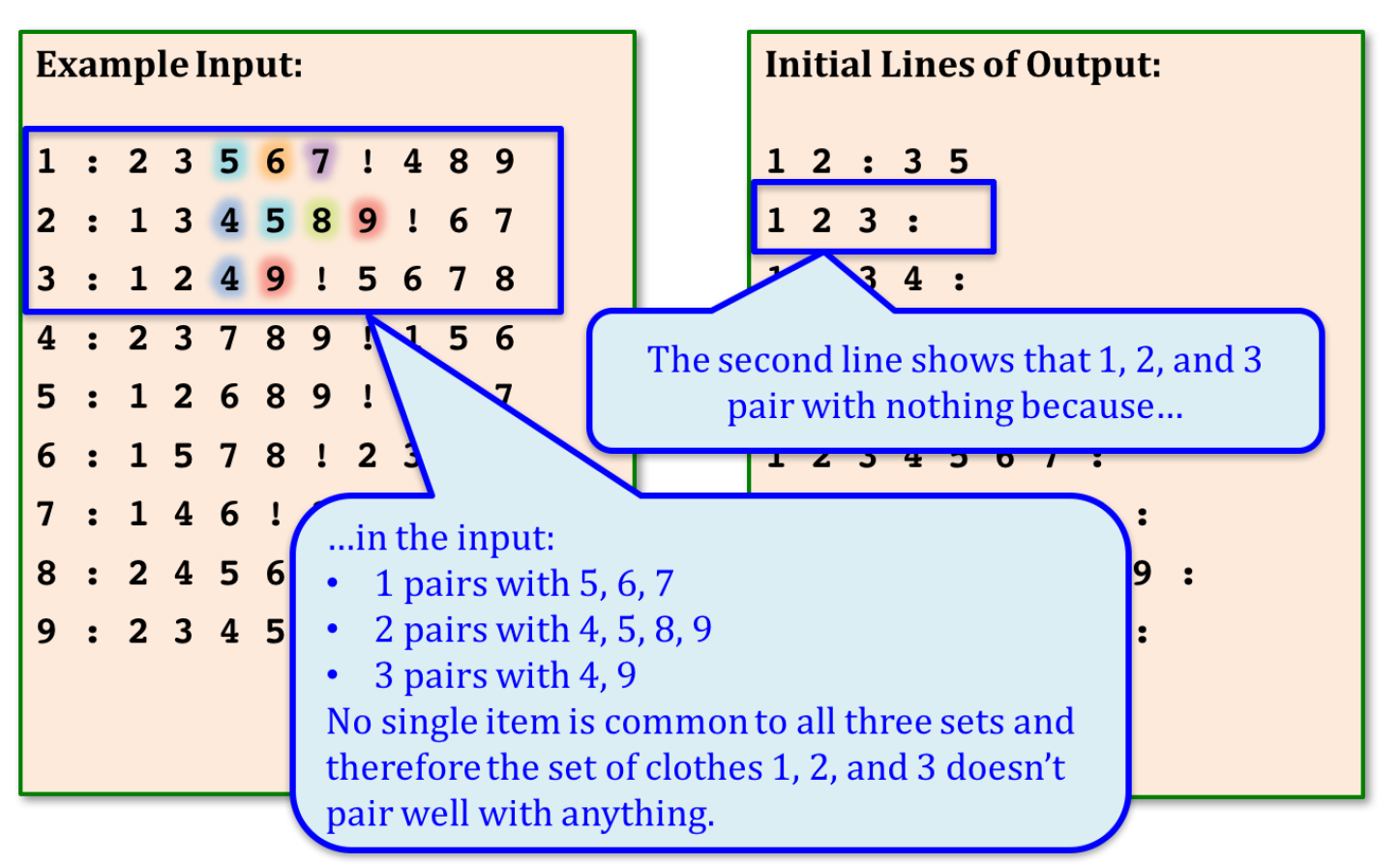


The first output line is saying that, if 1 and 2 are in the shopping cart, then a reasonable

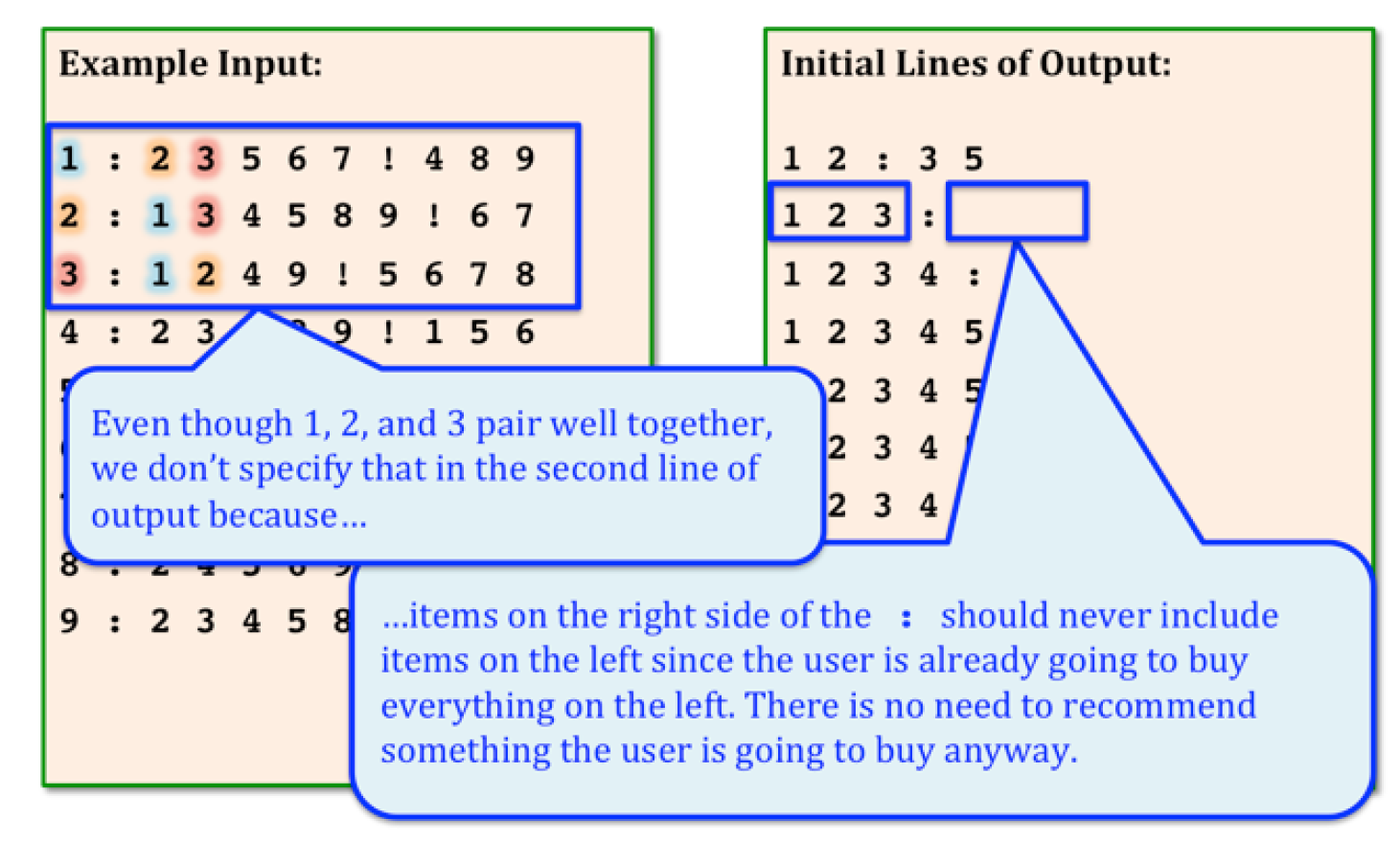
recommendation is to buy 3 and 5. Here’s why:



Now examine the second output line:

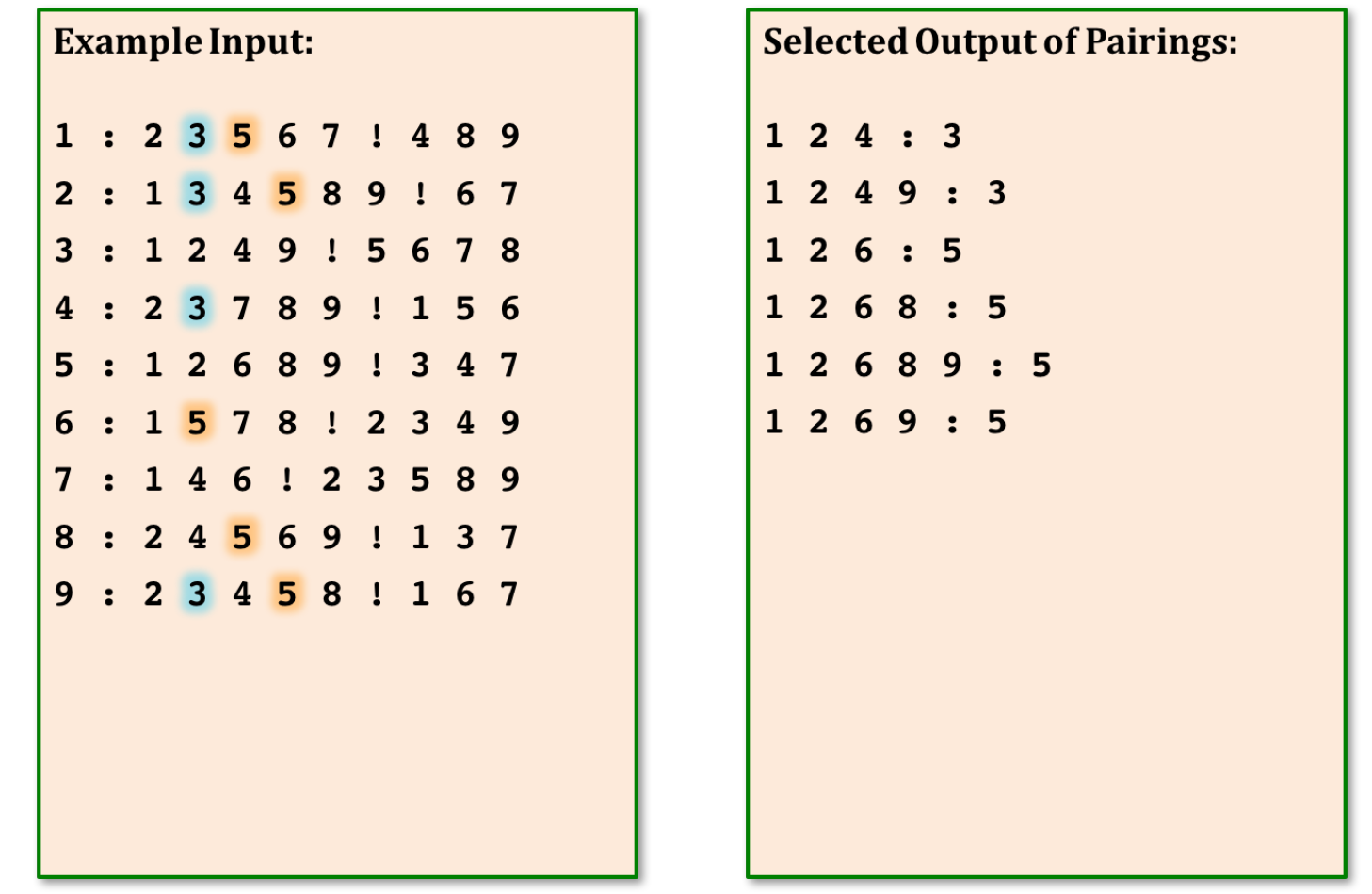


Notice that we don’t have to specify items the shopper is going to buy together anyway:



Since the full output is quite long, I will not be posting it here. Below, I’ve posted the next

several output lines where there is actually a pairing:



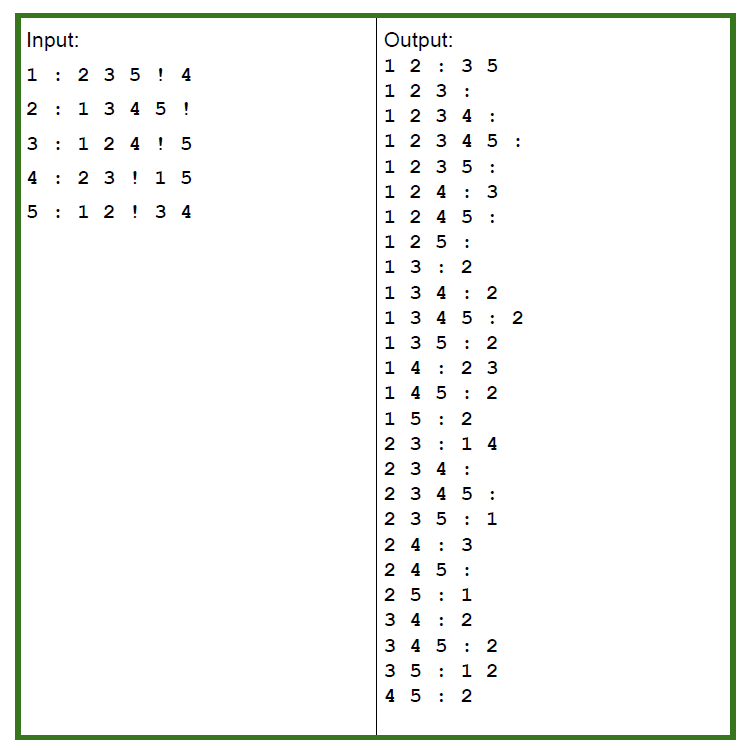
The combinations to the left of the colon in the output do not necessarily have to be in sorted

order as seen above (e.g. 1 2 4 coming before 1 2 4 7). However, *the values on either side of*

*the colon should be in sorted order* (e.g. 1 2 5 6 is correct, 1 5 2 6 is not). Here is a complete

example of input:

Complete Example



The left hand side should be a list of the clothing IDs separated by a space. This is followed by

a colon. The right hand side contains a sorted list of clothing IDs that pair well with everything

on the left hand side. These are also separated by a space.