WhiteSpace

WHITE SPACE CODING CHALLENGES

GAME RULES

AIM

This pack contains 3 coding challenges. Please choose 2 and complete the associated task(s).

TIMING

You have 2 weeks to complete the challenges. You are free to use any resource available to you.

DELIVERY

We will plan a report-out session 2 weeks after you've received this briefing pack. In the report-out you will be asked, for each challenge, to share your

- understanding of the challenge
- key assumptions and approach taken to tackle the challenge
- coding implementation and associated results

There is no need to prepare extensive presentation material. The session will take approximately 1 hour.

WhiteSpace

WHITE SPACE CODING CHALLENGE

TANGLED TRAFFIC

INTRODUCTION

You are given a map of a newly built town. The town has houses, shopping malls, and single city center – but no roads connecting them. Your assignment is to plan these roads.

There are two types of roads:

- *local* roads connect houses to houses, and houses to malls
- *express* roads connect malls to malls, and malls to the city center

CHALLENGE

Write an algorithm that

- takes as input
  - a set of N house coordinates (x,y)
  - a set of M mall coordinates (x,y)
  - the coordinates of the city center (x,y)
- returns
  - a list of planned local roads
  - a list of planned express roads
  - the total length of planned local roads
  - the total length of planned express roads
- visualizes the solution

RULES

Your algorithm must obey the following game rules:

- you must be able to travel from any house to any other house
- you must be able to travel from any house to at least one mall
- you must be able to travel from any house to the city center
- *travel* refers to traveling via local or express roads, possibly via other houses or malls
- not all malls need to be connected
- you may consider M and N to be of the order of 10

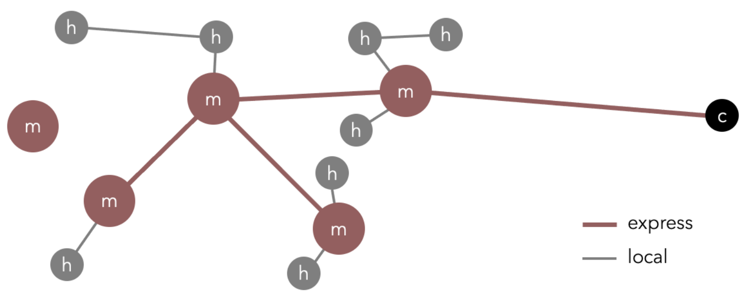

Figure: illustration of a road plan connecting houses (h), malls (m) and the city center (c) for N=8 and M=5.

OBJECTIVE

- Plan your roads such that the total road cost is minimized, with the total cost given by a weighted sum of the local and express total lengths, i.e. $cost = \alpha * total\_length_{local} + (1-\alpha) * total\_length_{express}$, with α variable and $\alpha \in [0,1]$.
- Discuss the edge cases $\alpha \to 0$ and $\alpha \to 1$.

BONUS

- Discuss the scaling of your algorithm

WhiteSpace

WHITE SPACE CODING CHALLENGE

BUSY BUSES

INTRODUCTION

Consider a bus on a route with N stops. At each stop there are passengers that wish to board. Like any real bus, this bus has a limited capacity, so there may be times when not all passengers that wish to board can fit on the bus. Your assignment is to determine which passengers are allowed to board at every stop.

CHALLENGE

Write an algorithm that
- takes as input
  - a number of stops N, with N not exceeding 20
  - a number of available bus seats C, with C not exceeding 30
  - a random list of passenger groups, each parameterized by $(group\_size, hop\_on\_location, hop\_off\_location)$, where
    - $group\_size$ is the number of passengers in the group, not exceeding 10
    - $hop\_on\_location$ indicates the group's start location,
    - $hop\_off\_location$ indicates the group's desired destination
- returns
  - a list of groups to board the bus
  - the total number of passengers that have boarded the bus
- visualizes the solution

RULES

Your algorithm must obey the following game rules:
- you cannot fit more people on the bus than its capacity allows for
- groups cannot be split
- you can assume locations are labelled alphabetically, and that groups only wish to visit locations that are alphabetically later than their starting position (i.e., groups can travel from A to C, but not from C to A)
- you can assume that, on average, there are more passengers willing to board than the bus capacity allows for

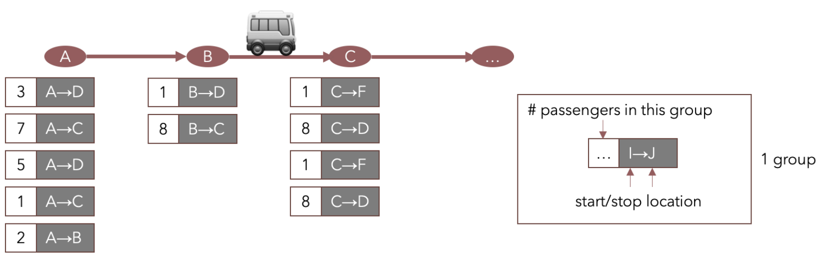

Figure: illustration of a series of bus stops (with 3 shown) with a series of groups wishing to board. There are 5 groups shown at location A, 2 at B and 4 at C. The first group at A consists of 3 people that wish to go to location D.

OBJECTIVES

- Assume that the price of a ticket (per person) is independent of the distance travelled (i.e., there is a fixed cost per person per ticket). Plan the allocation such that the *total income of the bus* is maximized.
- Alternatively, assume that the price of a ticket (per person) is directly proportional to the distance travelled. Plan the allocation such that the *total income of the bus* is maximized.

BONUS

- Discuss the scaling of your algorithm

WhiteSpace

WHITE SPACE CODING CHALLENGE

FASTEST FOOD

INTRODUCTION

You are head chef in a restaurant, and are given a list of orders coming in. Each order has a preparation time and a cooking time. Your assignment is to plan orders in time across preparation and cooking stations.

CHALLENGE

Write an algorithm that

- takes as input
  - a random list of orders of length N with their corresponding times (preparation and cooking)
- returns
  - the sequence in which orders must be prepared and cooked
- visualizes the solution

RULES

Your algorithm must obey the following game rules:

- You cannot prepare more than 2 orders at any given time
- You cannot cook more than 2 orders at any given time
- Orders can be either regular or kids' meals
- All orders must be prepared before they can be cooked
- Orders do not need to be cooked straight after being prepared
- Cooking times and preparation times are multiples of 5 minutes with a maximum of 1 hour

Order	Kids?	Prep time (minutes)	Cooking time (minutes)
1	Y	5	45
2	N	5	15
3	Y	10	30
4	N	35	55
5	N	5	60
6	N	20	15
7	N	10	5
…	…	…	…

Figure: illustration of a random realization of orders

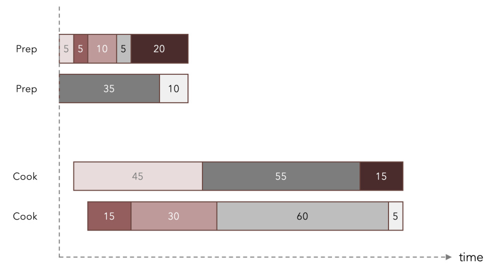

Figure: illustration of an example sequence

OBJECTIVE

- Plan the sequencing of orders such that the *total time needed to deliver (prepare and cook) all orders* is minimized.
- Assume you wish to prioritize kids' meals. Change your scoring function accordingly. Discuss if/how the total time to deliver all orders changes.
- How would you use the above results to decide on *how much* you are willing to prioritize kids' meals over regular ones?

BONUS

- Discuss the scaling of your algorithm