Assignment 3

In this assignment, we will design an algorithm that will give us the cheapest flights from source to destination. We will use real flight prices from kayak.com and develop our own DP algorithm that will give us a sequence of flights whose total cost is cheaper than the cheapest (one/multi-stop) flight on kayak.com

First, we will consider a smaller problem. Imagine that there are only 6 airports in the world and only 5 airlines.

NOTE: DO NOT CHANGE THE CODE HERE. ONLY FILL CODE IN FUNCTIONS WHERE IT IS ASKED.

```
airports <- c('BOM', 'NYC', 'DXB', 'LHR', 'FRA', 'DOH')
airlines <- c('AIR_INDIA', 'BRITISH_AIRWAYS', 'EMIRATES',
            'QATAR_AIRWAYS', 'LUFTHANSA')
```

Read data from csv files containing flight prices. Each csv is named after an airport. The prices in that csv correspond to prices for DIRECT FLIGHT, FROM that airport. The columns of the csv represent airline chosen and ROWS represent the DESTINATION

```
#setwd("/Users/gaurav/Personal/OSU/Spring18/CPDASP18/module_3/")

read_csv <- function(file_name) {
  temp <- read.csv(file_name)
  temp2 <- temp[,-1]
  rownames(temp2) <- temp$X
  temp2
}

BOM <- read_csv("BOM.csv")
NYC <- read_csv("NYC.csv")
DXB <- read_csv("DXB.csv")
LHR <- read_csv("LHR.csv")
FRA <- read_csv("FRA.csv")
DOH <- read_csv("DOH.csv")

price_matrix = list(BOM, NYC, DXB, LHR, FRA, DOH) # This is same order as airports
```

Q1: Write a function that returns the lowest cost of direct flight from BOM to NYC (3 points)

Fill the function below

```
lowest_cost_BOM_to_NYC_direct <- function() {
  # Write your code here

}

(lowest_cost_BOM_to_NYC_direct())
```

```
## NULL
```

Q2: Write a function that returns the lowest cost of direct flight from one airport to another (2 points)

Fill the function below

```
lowest_cost_direct_flight <- function(from, to) {
  # Write your code here
  # First get index of FROM airport to check which
  # data frame from price matrix to use
  # Since airports array and price_matrix has same order of airports
  index_of_from <- which(airports==from)[1]
  prices_from <- price_matrix[index_of_from][[1]]

  # Write your code here

}

(lowest_cost_direct_flight('BOM', 'NYC'))
```

```
##     AIR_INDIA BRITISH_AIRWAYS  EMIRATES QATAR_AIRWAYS LUFTHANSA
## BOM         0               0         0             0         0
## NYC      1300       149000000 149000000     149000000 149000000
## DXB       198       149000000       269     149000000 149000000
## LHR       641             598 149000000     149000000 149000000
## FRA 149000000       149000000 149000000     149000000      1371
## DOH      2326       149000000 149000000           925 149000000
```

Q3: Given an array of airports, write a function that outputs the lowest cost to travel from each airport in the array to any airport in the same array. The output should be an NxN matrix where N is length of array of airports. Note that diagonal elements will be 0 (5 points)

Fill the function below

```
lowest_cost_direct_flight_matrix <- function(airports) {
  # Write your code here
}

(lowest_cost_direct_flight_matrix(airports))
```

```
## NULL
```

Q4. Here comes the main question. Find the cheapest flight from any airport to any airport which may or maynot be direct flight. (6 points)

Fill the function below

```
lowest_cost_flight_matrix <- function(airports, max_layovers) {
  # Write your code here

}
```

Now lets check the lowest prices when max_layover is 1 and compare them with max_layover = 0 (direct flights).

```r
(lowest_cost_flight_matrix(airports,1))
```

```
## NULL
```

```r
(lowest_cost_flight_matrix(airports,0))
```

```
## NULL
```

Lets directly print a dataframe of dollars saved by increasing max_layover. Note that the optimal flight could also be a direct flight.

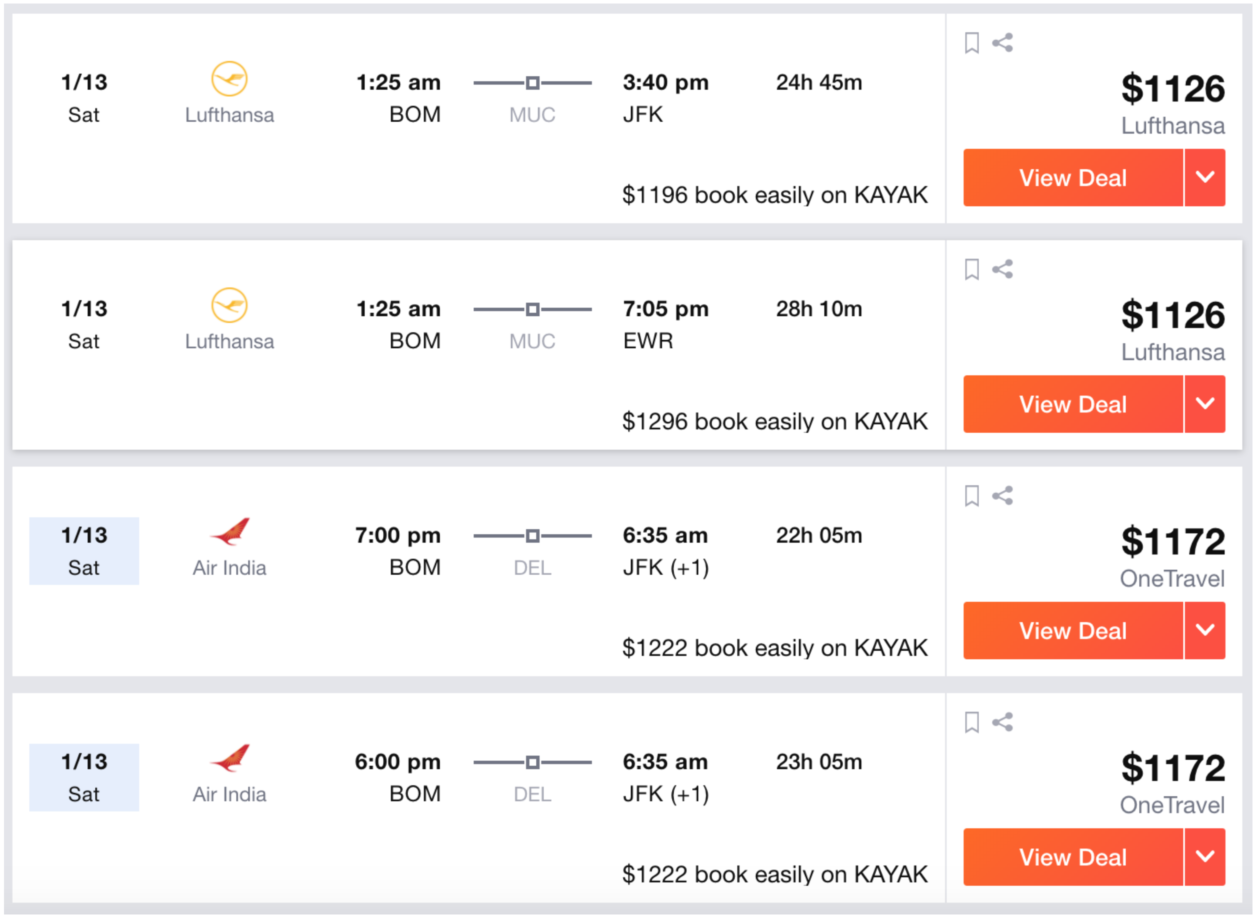
```r
(lowest_cost_flight_matrix(airports,0)-lowest_cost_flight_matrix(airports,1))
```

```
## numeric(0)
```

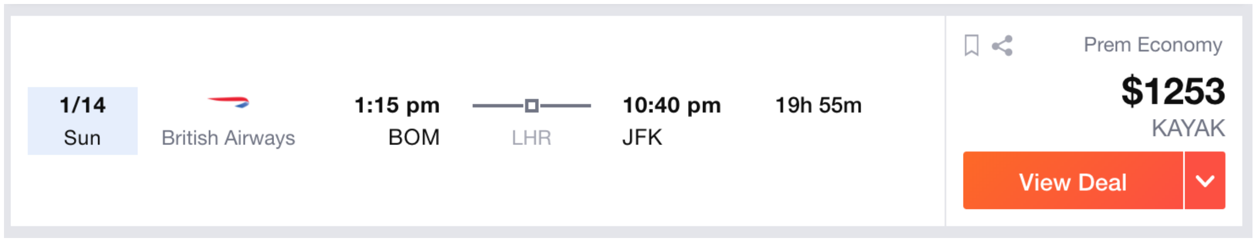
Note that the large numbers in dollars saved are because there was no direct flight but there were one stop flights, so technically you saved the cost of building and flying your own long range Boeing 747

We see that the lowest direct flight from BOM to NYC is $1300 (which is actual price on kayak.com) and one stop flight is $990. Lets see what kayak gives as the cheapest one stop flight of BOM to NYC for same dates.

We see that our algorithm gives much cheaper flights than online websites! Take BOM to LHR by BRITISH_AIRWAYS then take LHR to NYC by AIR_INDIA for a total of just $990.



1/13 Sat	Lufthansa	1:25 am BOM	MUC	3:40 pm JFK	24h 45m	$1196 book easily on KAYAK	$1126 Lufthansa
1/13 Sat	Lufthansa	1:25 am BOM	MUC	7:05 pm EWR	28h 10m	$1296 book easily on KAYAK	$1126 Lufthansa
1/13 Sat	Air India	7:00 pm BOM	DEL	6:35 am JFK (+1)	22h 05m	$1222 book easily on KAYAK	$1172 OneTravel
1/13 Sat	Air India	6:00 pm BOM	DEL	6:35 am JFK (+1)	23h 05m	$1222 book easily on KAYAK	$1172 OneTravel


1/14
Sun
British Airways
1:15 pm
BOM
LHR
10:40 pm
JFK
19h 55m
Prem Economy
$1253
KAYAK

Try changing max_layovers to 2. You will see a significant increase in runtime! The technique of memoization solves this (Memoization was demonstrated in python tutorial).

Q5. (Bonus Question) Try to use memoization

```
faster_lowest_cost_flight_matrix <- function(airports, max_layovers) {

}
```

Q6. (Bonus Question) What will happen if you try to increase number of states? Hint: Read curse of dimensionality in Dynamic Programming

```
faster_lowest_cost_flight_matrix <- function(airports, max_layovers) {

}
```

Now build your own website that offers cheapest flight tickets for patient customers that are willing to wait for their requests!