Introduction:

In this document, we present two versions of the analysis method used for our study. The first version contains the original code (Version 1), which served as the foundation for our analysis. It includes the required libraries, data reading, data preprocessing steps, and the primary analysis method used initially.

The second version showcases the modified code (Version 2) tailored to our specific dataset and research questions. We have made several adjustments to the original code to accommodate the unique features of our data and to address the specific objectives of our study. Each code block is preceded by adjustment notifications.

This comparative approach allows readers to gain insights into the adaptations made and facilitates transparency, reproducibility, and a deeper understanding of how the analysis method was tailored to suit our research needs.

Version 1 (Original Code):

Grid-sequence Analysis Tutorial

Overview

Grid-sequence analysis is a descriptive technique to capture within-dyad dynamics and allow for between-dyad comparisons. This analytic technique draws from state space grids, typically used in developmental psychology (Hollenstein, 2013) and sequence analysis, previously used in sociology and biology (Macindoe & Abbott, 2004).

Grid-sequence analysis combines state space grids and sequence analysis by:

1. Tracking a dyad's movements across a grid.
2. Converting dyadic movements into a univariate sequence.
3. Clustering dyads with similar movements using sequence analysis.

In this example, we will examine data simulated from couples' self-reported happiness collected six times a day for seven days using ecological momentary assessment. In this case, we have simulated 98 couples' data with 42 time points for each member of the couple. We are interested in whether couples' day-to-day emotion dynamics (specifically, happiness) are related to either couple members' relationship satisfaction or health (which have also been simulated).

Although the current example highlights how grid-sequence analysis can identify differences in variability across dyads, one particular advantage of grid-sequence analysis is identifying how use of the state space grid can change over time (e.g., certain states or patterns appearing early in the sequence and other states or patterns appearing later in the sequence).

In addition, the accompanying "GridSequenceAnalysis_Tutorial_2022August20.rmd" file contains all of the code presented in this tutorial and can be opened in RStudio (a somewhat more friendly user interface to R).

To read more about grid-sequence analysis see:

Brinberg, M., Fosco, G.M., & Ram, N. (2017). Examining inter-family differences in intra-family (parent-adolescent) dynamics using grid-sequence analysis. *Journal of Family Psychology, 31*(8), 994-1004. doi: 10.1037/fam0000371

Brinberg, M., Ram, N., Hülür, G., Brick, T.R., & Gerstorf, D. (2018). Analyzing dyadic data using grid-sequence analysis: Inter-dyad differences in intra-dyad dynamics. *Journals of Gerontology: Psychological Sciences, 73*(1), 5-18. doi: 10.1093/geronb/gbw160

Outline

In this tutorial, we'll cover...

- Reading in the data and loading needed packages.
- Creating state space grids for each dyad.
- Creating sequences.
- Establishing a cost matrix and conducting sequence analysis.
- Determining the number of clusters.
- Examining group differences among clusters.
- Cautions.
- Conclusions.

Read in the data and load needed packages.

Let's read the data into R.

We are working with two data sets in this tutorial. One data set contains repeated measures ("gridsequence_simulation_data"), specifically the simulated reports of happiness for each dyad member. The other data set contains person-level data ("gridsequence_simulation_persons"), specifically the simulated reports of relationship satisfaction and health for each dyad member.

Both data sets are stored as .csv files (comma-separated values file, which can be created by saving an Excel file as a csv document) on my computer's desktop.

```
# Set working directory (i.e., where your data file is stored)
# This can be done by going to the top bar of RStudio and selecting
# "Session" --> "Set Working Directory" --> "Choose Directory" -->
# finding the location of your file
setwd("~/Desktop") # Note: You can skip this line if you have
#the data files and this .rmd file stored in the same directory

# Read in the repeated measures data
data <- read.csv(file = "gridsequence_simulation_data.csv", head = TRUE, sep = ",")

# View the first 10 rows of the repeated measures data
head(data, 10)

# Read in the person-level data
persons <- read.csv(file = "gridsequence_simulation_persons.csv", head = TRUE, sep = ",")

# View the first 10 rows of the persons data
head(persons, 10)
```

In the repeated measures data ("data"), we can see each row contains information for one time point and there are multiple time points for each dyad member within each dyad. In this data set, there are columns for:

- Dyad ID (`id`)
- Time variable - in this case, a momentary assessment (`time`)
- Repeated measures variable for one dyad member - in this case, simulated self-reports of happiness from dyad member 1 (`outcome1`)
- Repeated measures variable for the other dyad member - in this case, simulated self-reports of happiness from dyad member 2 (`outcome2`)

In the person-level data ("persons"), we can see there is one row for each dyad member and there are columns for:

- Dyad ID (`id`)
- Dyad member (`member`)
- Person-level variables - in this case, simulated dyad members' relationship satisfaction (`rel_sat`) and health status (`health`)

Load the R packages we need.

Packages in R are a collection of functions (and their documentation/explanations) that enable us to conduct particular tasks, such as plotting or fitting a statistical model.

```
# install.packages("cluster")          # Install package if you have never used it before
library(cluster)                       # For cluster analysis

# install.packages("devtools")         # Install package if you have never used it before
require(devtools)                      # For version control

# install.packages("ggplot2")          # Install package if you have never used it before
library(ggplot2)                       # For plotting

# install.packages("reshape")          # Install package if you have never used it before
library(reshape)                       # For reshaping the data (long to wide)

# install.packages("TraMineR")         # Install package if you have never used it before
library(TraMineR)                      # For sequence analysis

# install.packages("TraMineRextras")   # Install package if you have never used it before
library(TraMineRextras)                # For sequence analysis
```

Create State Space Grids for Each Dyad.

The first step in grid-sequence analysis is to establish the state space grid that will collapse the dyads' bivariate time series into a univariate categorical time series.

We create the state space grid by setting up cut points. In this example, we create a 4 x 4 (16 cell) grid, and use three cut points (at intervals of 25 because our measure of happiness is on a 0-100 scale). Our cut points are only for the sake of demonstration, but these divisions can vary depending on your research question or underlying theory.

```
# Create cut points by creating new variables within the repeated measures data set ("data")
data$cut1 <- 25
data$cut2 <- 50
data$cut3 <- 75
```

Next, we use these cut points to label the cells on the grid with letters of the English alphabet and create a new variable that contains these letters (labeled as "happy" in our data set "data"). The lettering of the cells is a simple way to create categories that represent information from both dyad members in a univariate space. This new variable will be used in the subsequent sequence analysis.

To label the state space grid, we set the x-axis to be dyad member 1 (who is associated with "outcome1") and the y-axis to be dyad member 2 (who is associated with "outcome2"). In this case, we label the upper-left hand corner with "A" and the lower-right hand corner with "P," filling in alphabetically along the way.

```
# Create new variable "happy" that will contain cell labels
data$happy <- NA

# Letter each cell of the grid

# Set the value of "happy" to "A" when the value of "outcome1" is less than cut-point 1 and when the va
# Similar logic follows for the labeling of all the other cells below
data$happy[data$outcome1 <= data$cut1 &
           data$outcome2 > data$cut3] <- "A"

data$happy[data$outcome1 > data$cut1 &
           data$outcome1 <= data$cut2 &
           data$outcome2 > data$cut3] <- "B"

data$happy[data$outcome1 > data$cut2 &
           data$outcome1 <= data$cut3 &
           data$outcome2 > data$cut3] <- "C"

data$happy[data$outcome1 > data$cut3 &
           data$outcome2 > data$cut3] <- "D"

data$happy[data$outcome1 <= data$cut1 &
           data$outcome2 > data$cut2 &
           data$outcome2 <= data$cut3] <- "E"

data$happy[data$outcome1 > data$cut1 &
           data$outcome1 <= data$cut2 &
           data$outcome2 > data$cut2 &
           data$outcome2 <= data$cut3] <- "F"

data$happy[data$outcome1 > data$cut2 &
           data$outcome1 <= data$cut3 &
           data$outcome2 > data$cut2 &
           data$outcome2 <= data$cut3] <- "G"

data$happy[data$outcome1 > data$cut3 &
           data$outcome2 > data$cut2 &
           data$outcome2 <= data$cut3] <- "H"

data$happy[data$outcome1 <= data$cut1 &
```

```
           data$outcome2 > data$cut1 &
           data$outcome2 <= data$cut2] <- "I"

data$happy[data$outcome1 > data$cut1 &
           data$outcome1 <= data$cut2 &
           data$outcome2 > data$cut1 &
           data$outcome2 <= data$cut2] <- "J"

data$happy[data$outcome1 > data$cut2 &
           data$outcome1 <= data$cut3 &
           data$outcome2 > data$cut1 &
           data$outcome2 <= data$cut2] <- "K"

data$happy[data$outcome1 > data$cut3 &
           data$outcome2 > data$cut1 &
           data$outcome2 <= data$cut2] <- "L"

data$happy[data$outcome1 <= data$cut1 &
           data$outcome2 <= data$cut1] <- "M"

data$happy[data$outcome1 > data$cut1 &
           data$outcome1 <= data$cut2 &
           data$outcome2 <= data$cut1] <- "N"

data$happy[data$outcome1 > data$cut2 &
           data$outcome1 <= data$cut3 &
           data$outcome2 <= data$cut1] <- "O"

data$happy[data$outcome1 > data$cut3 &
           data$outcome2 <= data$cut1] <- "P"

# View the first 10 rows of the data
head(data, 10)
```

It is useful to plot each dyad's movements across the state space grid to obtain a greater understanding of the data. Here are plots of two different dyads (one with low variability and one with high variability).

Example Dyad 1: Low variability.

Example Dyad 2: High variability.

We will create a loop that plots each dyad in the data set and saves these plots to a PDF.

First, identify the location where you would like to save the PDF and save this location as the object "dir".

```
# This can be done by going to the top bar of RStudio and selecting
# "Session" --> "Set Working Directory" --> "Choose Directory" -->
# finding the location of when you want your file
dir <- setwd("~/Desktop")
```

Second, create a vector of all the IDs in the data set.

```
# Create vector
idlist <- unique(data$id)
```

```
# View contents of vector
idlist
```

Finally, create and run the loop for the plots.

```
# Open the pdf file
pdf('State Space Grid Simulation.pdf', width = 10, height = 7)

for(x in 1:length(idlist)) #looping through plots
  {

    # Select participant ID from the vector of IDs
    subject_id <- idlist[x]

    # Subset data for selected participant ID
    data_sub <- subset(data, id == subject_id)

    # Create object with participant ID
    name <- as.character(data_sub$id[1])

    # Remove the missing data
    # So only points will be plotted when there is information
    # from both dyad members
    data_sub_noNA <- na.omit(data_sub)

    if(nrow(data_sub_noNA) == 0){}

    if(nrow(data_sub_noNA) > 0){


  grid_plot <-

  ggplot(# Select data and set variables of x- and y- axes
         data = data_sub_noNA, aes(x = outcome1, y = outcome2)) +

         # Set x-axis limits
         # Give a little extra room (the scale is 0-100) so points on the
         # edge of the scale will show up easily
         xlim(-5, 105) +

         # Set y-axis limits
         # Give a little extra room (the scale is 0-100) so points on the
         # edge of the scale will show up easily
         ylim(-5, 105) +

         # Label y-axis
         ylab('Happiness Partner 2') +

         # Label x-axis
         xlab('Happiness Partner 1') +

         # Create each cell of the state space grid by setting its width (xmin and xmax),
         # height (ymin and ymax), and color (fill using hex code: https://www.color-hex.com/)
         # Set transparency of cell with the "alpha" command
```

```
        geom_rect(xmin = 0, xmax = cut1, ymin = 0, ymax = cut1, fill = "#000000", alpha = .05) +
        geom_rect(xmin = 0, xmax = cut1, ymin = cut1, ymax = cut2, fill ="#000054", alpha = .05) +
        geom_rect(xmin = 0, xmax=cut1, ymin = cut2, ymax = cut3, fill = "#0000A8", alpha = .05) +
        geom_rect(xmin = 0, xmax=cut1, ymin = cut3, ymax = 100, fill = "#0000FC", alpha = .05) +
        geom_rect(xmin = cut1, xmax = cut2, ymin = 0, ymax = cut1, fill = "#540000", alpha = .05) +
        geom_rect(xmin = cut1, xmax = cut2, ymin = cut1, ymax = cut2, fill = "#540054", alpha = .05) +
        geom_rect(xmin = cut1, xmax = cut2, ymin = cut2, ymax = cut3, fill = "#5400A8", alpha = .05) +
        geom_rect(xmin = cut2, xmax = cut3, ymin = 0, ymax = cut1, fill = "#A80000", alpha = .05) +
        geom_rect(xmin = cut2, xmax = cut3, ymin = cut1, ymax = cut2, fill = "#A80054", alpha = .05) +
        geom_rect(xmin = cut2, xmax = cut3, ymin = cut2, ymax = cut3, fill = "#A800A8", alpha = .05) +
        geom_rect(xmin = cut2, xmax = cut3, ymin = cut3, ymax = 100, fill = "#A800FC", alpha = .05) +
        geom_rect(xmin = cut3, xmax = 100, ymin = 0, ymax = cut1, fill = "#FC0000", alpha = .05) +
        geom_rect(xmin = cut3, xmax = 100, ymin = cut2, ymax = cut3, fill = "#FC00A8", alpha = .05) +
        geom_rect(xmin = cut3, xmax = 100, ymin = cut3, ymax = 100, fill = "#FC00FC", alpha = .05) +

        # Additional plot aesthetics
        theme(
               # Adjust text size and color on x-axis and y-axis
               axis.text = element_text(size = 14, color = 'black'),

               # Adjust size of plot title
               axis.title = element_text(size = 18),

               # Adjust color and lines surrounding plot
               panel.grid.major = element_line(colour = "white"),
               panel.grid.minor = element_blank(),
               panel.background = element_blank(),
               axis.ticks = element_blank()) +

         # Vertical lines to divide grid
         geom_vline(xintercept = c(0,25,50,75,100)) +

         # Horizontal lines to divide grid
         geom_hline(yintercept = c(0,25,50,75,100)) +

         # Create point that represents the x-y intersection point of the dyads' variables
         geom_point(colour = "white") +

         # Create line that connects the consecutive points in time
         geom_path(colour = "white")  +

         # Create title for plot by combining "Dyad = " with the name object created above
         ggtitle(paste("Dyad =", name))

  print(grid_plot)
}

dev.off()
```

Examination of dyads' state space grids can provide valuable insights into within-dyad dynamics and how these dynamics differ from dyad-to-dyad.

Create Sequences.

In this step, we:

(1) reformat the repeated measures data ("data") from long to wide,

(2) create an "alphabet" that represents each of our categories, and

(3) create and plot the categorical sequence data.

1. Reformat the data from long to wide.

Our repeated measures data ("data") are currently in a "long" format, which means that each dyad has multiple rows with each assessment having its own row. For sequence analysis, we need our repeated measures data to be in a "wide" format, which means that each dyad should only have one row and each assessment of the study should have its own column.

```
# Reformat repeated measures data ("data") from long to wide
data_wide <- reshape(
                     # Select data and specific columns that should be kept
                     # (dyad ID: id, time variable: time,
                     # categorical variable: happy)
                     data = data[, c("id", "time", "happy")],
                     # Identify time variable ("time")
                     timevar = "time",
                     # Identify dyad ID variable ("id")
                     idvar = "id",
                     # Identify categorical variable ("happy")
                     v.names = "happy",
                     # Note direction of data reformat
                     direction = "wide",
                     # No spaces in new column names
                     sep = "")

# View the first 10 rows and first 10 columns of the wide data
head(data_wide[1:10, 1:10])
```

2. Create alphabet.

We create an alphabet that represents each possible category within the categorical variable of interest (in this case, "happy"). The actual naming of these values is not important, but the categories can be named in a way that facilitates interpretation.

```
# Create a vector that contains the categories that appear in the wide data set
gs_alphabet <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P")

# Create a vector that allows for more helpful labels if applicable (e.g., negative, neutral, and posit
gs_labels <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P")
```

3. Create sequences.

Now that the data are in the correct (i.e., wide) format and we have created an alphabet for our sequence analysis, we next need to create a sequence object that can be understood by the R package for sequence analysis (`TraMineR`).

Before creating the sequences, we first we assign colors to each of the categories, which will help us when viewing plots of the sequences. This step is not required since there is a default color palette, but this gives us control over what color is assigned to which category. We do this by assigning each category (i.e., turn type) a hex code (https://www.color-hex.com/). The categories should be written as they appear in the alphabet created above.

A note on accessibility: To make your plots accessible, you may consider adopting a colorblind-friendly palette. David Nichols' website (https://davidmathlogic.com/colorblind/) provides a great explainer on this issue, as well as a color picking tool.

```
A <- "#0000FC"
B <- "#5400FC"
C <- "#A800FC"
D <- "#FC00FC"
E <- "#0000A8"
F <- "#5400A8"
G <- "#A800A8"
H <- "#FC00A8"
I <- "#000054"
J <- "#540054"
K <- "#A80054"
L <- "#FC0054"
M <- "#000000"
N <- "#540000"
O <- "#A80000"
P <- "#FC0000"
```

Next, we create an object ("happy_seq") that contains all of the sequences in the format needed for the sequence analysis package.

```
happy_seq <- seqdef(data_wide,                          # Select data
                    var = 2:41,                         # Columns containing repeated measures data
                    alphabet = gs_alphabet,             # Alphabet
                    labels = gs_labels,                 # Labels
                    xtstep = 6,                         # Steps between tick marks
                    cpal=c(A, B, C, D, E, F, G, H,
                           I, J, K, L, M, N, O, P))# Color palette
```

A lot of text will appear after the sequence object is created. This text tells you about the number of sequences (which should be equal to the number of dyads in the sample), the states (i.e., categories) that appear in the sequence object, and the alphabet and labels of the categories.

Finally, we can plot the sequences.

```
seqIplot(happy_seq,                     # Sequence object
         with.legend = "right",         # Display legend on right side of plot
         cex.legend = 0.8,              # Change size of legend
         main = "Dyad Happiness")       # Plot title
```

To read this plot, each row represents one dyad's state space grid location throughout the study and each color represents a different state space grid cell. We can see that the location and movement throughout the state space grid varied considerably from dyad to dyad.

Establish a Cost Matrix and Sequence Analysis.

There is one last step before conducting sequence analysis: establishing a cost matrix.

Sequence analysis aims to minimize the "cost" of transforming one sequence into another and relies on an optimal matching algorithm. There are costs for inserting, deleting, and substituting letters, as well as costs for missingness. The output of sequence analysis is a n x n (n = number of dyads) dissimilarity matrix with the cost of transforming one sequence into the corresponding sequence in each cell of the matrix.

The researcher establishes a cost matrix and often use standards, such as insertion/deletion costs of 1.0 and missingness costs of half the highest cost, within the matrix. There are a number of ways to determine substitution cost, typically, cost is established as the distance between cells. In this case, we use Manhattan (city-block) distance to determine substitution costs, but other methods (e.g., Euclidian distance) are also possible.

There may be instances when it does not make sense to order categories. For the sake of demonstration, we show how to obtain a constant cost matrix below.

Example constant cost matrix - i.e., the cost of substituting one category for another is the same regardless of the category (this may be a useful approach if working with categories that are truly categorical and not ordinal).

```
# Create substitution cost matrix and save to the object "costmatrix"
costmatrix <- seqsubm(happy_seq,            # Sequence object
                      method = "CONSTANT",  # Method to determine costs
                      cval = 2,             # Substitution cost
                      with.missing = TRUE,  # Allows for missingness state
                      miss.cost = 1,        # Cost for substituting a missing state
                      time.varying = FALSE, # Does not allow the cost to vary over time
                      weighted = TRUE)      # Allows weights to be used when applicable

# Examine substitution cost matrix
costmatrix
```

Here, we will create our own cost matrix since we believe there are differential costs for substituting categories within the sequences (e.g., a lower cost to substitute categories that are near each other on the state space grid versus a higher cost to substitute categories that are far from each other on the state space grid).

The cost matrix will be a (k+1) by (k+1) with k = number of cells in the grid. However, because we have no missing data, the cost matrix will be n by n. When there is missing data, an additional column and row would be needed such that the right-most column and the bottom row would represent missingness costs (half of the highest cost, which in this case is half of 6).

We used Manhattan (city-block) distance to calculate costs. We calculated Manhattan distance manually and then inserted the results into the matrix below. For example, the cost of staying in the same cell is zero, the cost of moving from "A" to "B" is one, and the cost of moving from "A" to "N" is four.

Create new cost matrix.

```
costmatrix2 <- matrix(c(0.0, 1.0, 2.0, 3.0, 1.0, 2.0, 3.0, 4.0, 2.0, 3.0, 4.0, 5.0, 3.0, 4.0, 5.0, 6.0,
                        1.0, 0.0, 1.0, 2.0, 2.0, 1.0, 2.0, 3.0, 3.0, 2.0, 3.0, 4.0, 4.0, 3.0, 4.0, 5.0,
                        2.0, 1.0, 0.0, 1.0, 3.0, 2.0, 1.0, 2.0, 4.0, 3.0, 2.0, 3.0, 5.0, 4.0, 3.0, 4.0,
                        3.0, 2.0, 1.0, 0.0, 4.0, 3.0, 2.0, 1.0, 5.0, 4.0, 3.0, 2.0, 6.0, 5.0, 4.0, 3.0,
                        1.0, 2.0, 3.0, 4.0, 0.0, 1.0, 2.0, 3.0, 1.0, 2.0, 3.0, 4.0, 2.0, 3.0, 4.0, 5.0,
                        2.0, 1.0, 2.0, 3.0, 1.0, 0.0, 1.0, 2.0, 2.0, 1.0, 2.0, 3.0, 3.0, 2.0, 3.0, 4.0,
                        3.0, 2.0, 1.0, 2.0, 2.0, 1.0, 0.0, 1.0, 3.0, 2.0, 1.0, 2.0, 4.0, 3.0, 2.0, 3.0,
                        4.0, 3.0, 2.0, 1.0, 3.0, 2.0, 1.0, 0.0, 4.0, 3.0, 2.0, 1.0, 5.0, 4.0, 3.0, 2.0,
```

```
                    2.0, 3.0, 4.0, 5.0, 1.0, 2.0, 3.0, 4.0, 0.0, 1.0, 2.0, 3.0, 1.0, 2.0, 3.0, 4.0,
                    3.0, 2.0, 3.0, 4.0, 2.0, 1.0, 2.0, 3.0, 1.0, 0.0, 1.0, 2.0, 2.0, 1.0, 2.0, 3.0,
                    4.0, 3.0, 2.0, 3.0, 3.0, 2.0, 1.0, 2.0, 2.0, 1.0, 0.0, 1.0, 3.0, 2.0, 1.0, 2.0,
                    5.0, 4.0, 3.0, 2.0, 4.0, 3.0, 2.0, 1.0, 3.0, 2.0, 1.0, 0.0, 4.0, 3.0, 2.0, 1.0,
                    3.0, 4.0, 5.0, 6.0, 2.0, 3.0, 4.0, 5.0, 1.0, 2.0, 3.0, 4.0, 0.0, 1.0, 2.0, 3.0,
                    4.0, 3.0, 4.0, 5.0, 3.0, 2.0, 3.0, 4.0, 2.0, 1.0, 2.0, 3.0, 1.0, 0.0, 1.0, 2.0,
                    5.0, 4.0, 3.0, 4.0, 4.0, 3.0, 2.0, 3.0, 3.0, 2.0, 1.0, 2.0, 2.0, 1.0, 0.0, 1.0,
                    6.0, 5.0, 4.0, 3.0, 5.0, 4.0, 3.0, 2.0, 4.0, 3.0, 2.0, 1.0, 3.0, 2.0, 1.0, 0.0),
                    nrow = 16, ncol = 16, byrow=TRUE)

# Fix the row and column names to nice labels from first cost matrix
rownames(costmatrix2) <- rownames(costmatrix)
colnames(costmatrix2) <- colnames(costmatrix)
costmatrix2
```

Now that we have created the substitution cost matrix, we can conduct our sequence analysis.

We use an optimal matching algorithm for sequence analysis. The output of the sequence analysis is a n x n (n = number of dyads) dissimilarity matrix with the cost of transforming one sequence into every other sequence in the corresponding cell of the matrix.

Note: Other algorithms are available, and they can be specified in the method = " " argument below. To see other algorithms available in the TraMineR package, type ?seqdist in the console or type seqdist in the search bar at the top of the Help tab on the right.

```
# Conduct sequence analysis
dist_om <- seqdist(happy_seq,              # Sequence object
                   method = "OM",          # Optimal matching algorithm
                   indel = 1,              # Insert/deletion costs set to 1
                   sm = costmatrix2,       # Substitution cost matrix
                   with.missing = TRUE)

# Examine the top left corner of the dissimilarity matrix
dist_om[1:10, 1:10]
```

Cluster Determination.

We next take the distance matrix obtained in the prior step to determine an appropriate number of clusters that represent the different emotion dynamic patterns within our dyads. Several clustering techniques are available, but we use hierarchical cluster analysis using Ward's single linkage method. Other possible methods include k-medoids clustering or latent mixture models. After determining the number of clusters that work for the data, we create an object that contains cluster membership for each dyad (which will be used in the final step) and plot the clusters.

Conduct hierarchical cluster analysis and save cluster analysis results to the object "clusterward".

```
# Insert dissimilarity matrix ("dist_om"),
# indicate that we are using a dissimilarity matrix, and
# indicate that we want to use Ward's single linkage clustering method
clusterward <- agnes(dist_om, diss = TRUE, method = "ward")

# Plot the results of the cluster analysis using a dendrogram
# Insert cluster analysis results object ("clusterward")
plot(clusterward, which.plot = 2)
```

In this example, the resulting dendrogram indicates three clusters. We reached this conclusion by examining the length of the vertical lines (longer vertical lines indicate greater differences between groups) and the number of dyads within each group (we didn't want a group with too few dyads). After selecting a three cluster solution, we plotted the sequences of the three clusters for visual comparison.

```
# Cut dendrogram (or tree) by the number of
# determined groups (in this case, 3)
# Insert cluster analysis results object ("clusterward")
# and the number of cut points
cl3 <- cutree(clusterward, k = 3)

# Turn cut points into a factor variable and label them
# Insert cut point object ("cl3") and
# create labels by combining the text "Type" with either 1, 2, or 3
cl3fac <- factor(cl3, labels = paste("Type", 1:3))

# Plot the sequences for each cluster
seqplot(happy_seq,              # Sequence object
        group = cl3fac,         # Grouping factor level variable
        type = "I",             # Create whole sequence plot
        sortv = "from.start",   # Sort sequences based upon the category in which they begin
        with.legend = "right", # Display legend on right side of plot, change to "auto" if overlapping
        cex.legend = 0.8,       # Change size of legend
        border = NA)            # No plot border
```

It appears that "Type 1" dyads are composed of dyads in which both members are generally happy (as indicated by the bright pink colors that are in the top-right of the state space grid). "Type 2" dyads seem to be composed of dyads in which dyad members experience low levels of happiness (as indicated by the dark colors that are in the bottom-left of the state space grid). Finally, "Type 3" dyads appear to exhibit high variability in their levels of happiness.

In the next (and final) step, we will formally test whether the dyads within these clusters differ on support self-reported relationship satisfaction and health.

Examine Group Differences among Clusters.

The final step of of our analysis is to examine group differences among the clusters. One can use a variety of methods to examine group differences, and the choice of method will depend on the number of clusters chosen and the research question. For example, if two clusters are chosen and one wants to examine differences on an outcome variable between the clusters, then one would use a t-test. In this case, we use analysis of variance (ANOVA) to examine group differences.

We examined whether dyad members' self-reported relationship satisfaction and health (both simulated variables) differed by cluster membership, which represented daily dynamics of happiness.

```
# Add grouping variable to persons data set
persons$cl3 <- cl3

# Run ANOVA

rel_sat <- aov(persons$rel_sat ~ factor(persons$cl3))
summary(rel_sat)
# If post hoc test is needed
```

```
TukeyHSD(rel_sat)

health <- aov(persons$health ~ factor(persons$cl3))
summary(health)
# If post hoc test is needed
TukeyHSD(health)
```

As you can see, there were no significant group differences with this simulation data. Alternatively, we could examined dyad members separately (e.g., if dyad member 1 = male partners, dyad member 2 = female partners) to assess whether relationship satisfaction and subjective health differed by cluster membership for each dyad member type.

Cautions.

Although there are several distinct advantages of grid-sequence analysis, there are several limitations and considerations to the process, which include:

1. The need for simultaneous data from each member of the dyad.
2. The length of time series needed (dependent on the process under examination, but could be lengthy).
3. The change of a continuous variable into an interval variable when creating axes.
4. The extent of missingness.
5. The (current) need to use distinguishable dyads, although this can be flexible if the state space grid is symmetric across the bottom-left to top-right diagonal.

Conclusion.

Theories of interpersonal dynamics and social interaction emphasize the need to study within-dyad dynamics. Grid-sequence analysis is a flexible new approach that allows researchers to capture within-dyad dynamics and to make between-dyad comparisons using repeated-measures dyadic data. We hope that the combination of state space grids and sequence analysis help researchers better study and understand the processes of interpersonal interactions.

Version 2 (Modified Code):

0. Introduction to Grid-Sequence Analysis.

```
#loading libraries
library(cluster)
library(ggplot2)
library(reshape)
library(TraMineR)
```

```
##
## TraMineR stable version 2.2-6 (Built: 2022-12-13)
```

```
## Website: http://traminer.unige.ch

## Please type 'citation("TraMineR")' for citation information.
```

```
library(TraMineRextras)
```

```
## Warning: Paket 'TraMineRextras' wurde unter R Version 4.2.3 erstellt

## TraMineRextras stable version 0.6.6 (Built: 2023-03-23)

## Functions provided by this package are still in test

##     and subject to changes in future releases.
```

Data: Aggression is measured through repeated measures of monetary selection during the interactive Taylor Aggression Paradigm (iTAP).

```
#filepath for repeated measures if file located in the same directory as your R script
filepath <- "GSA_indis.csv"
data_TAP <- read.csv(filepath,header=TRUE)
head(data_TAP)
```

```
##   X id time A  B
## 1 1  1    1 0  0
## 2 2  1    2 0 10
## 3 3  1    3 0  0
## 4 4  1    4 0  0
## 5 5  1    5 0  0
## 6 6  1    6 0  0
```

```
#delete first column of each data set (1, 2, ..., n labeling in Excel sheet)
data_TAP[1] <- NULL
head(data_TAP)
```

```
##   id time A  B
## 1  1    1 0  0
## 2  1    2 0 10
## 3  1    3 0  0
## 4  1    4 0  0
## 5  1    5 0  0
## 6  1    6 0  0
```

Create an indistinguishable version of dataset by sorting dyad members in a way that only allows one direction of aggression score combinations: Per trial, the higher aggression score between both members is placed on the x-axis, and the lower aggression score is placed on the y-axis, resulting in scores occupying only the upper triangle of the state space grid.

```
# Initialize vectors x and y
x <- numeric(nrow(data_TAP))
y <- numeric(nrow(data_TAP))


# Compare values in columns A and B and assign to vectors x and y
x <- pmin(data_TAP$A, data_TAP$B)
y <- pmax(data_TAP$A, data_TAP$B)

# Create a new data frame 'ind_TAP' by combining columns

ind_TAP <- data.frame(id = data_TAP$id,
                      time = data_TAP$time,
                      x = x,
                      y = y)
```

We utilized the complete range of our measurement scale, spanning from 0 to 90 in increments of 10, to establish cut points. As a result, a 10 x 10 grid with 100 cells was created.

```
#Creating cut points
data_TAP$cut1 <- 0
data_TAP$cut2 <- 10
data_TAP$cut3 <- 20
data_TAP$cut4 <- 30
data_TAP$cut5 <- 40
data_TAP$cut6 <- 50
data_TAP$cut7 <- 60
data_TAP$cut8 <- 70
data_TAP$cut9 <- 80
data_TAP$cut10 <- 90
```

We added the new variable (labeled 'aggression') capturing the dyad-level time series to the dataset. For indistinguishable dyads, we implemented a mirrored mapping of the state space grid cells across the diagonal of the grid. This means that cells in the upper-left-hand corner and the lower-right-hand corner of the grid are both labeled with the same category, denoted as 'A'.

```
#Lettering each cell of the grid

data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut10] <- "A"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut10] <- "B"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut10] <- "C"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut10] <- "D"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut10] <- "E"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut10] <- "F"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut10] <- "G"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut10] <- "H"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut10] <- "I"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut10] <- "J"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut9] <- "K"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut9] <- "L"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut9] <- "M"
```

```
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut9] <- "N"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut9] <- "O"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut9] <- "P"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut9] <- "Q"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut9] <- "R"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut9] <- "S"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut9] <- "I"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut8] <- "U"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut8] <- "V"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut8] <- "W"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut8] <- "X"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut8] <- "Y"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut8] <- "Z"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut8] <- "AA"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut8] <- "AB"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut8] <- "R"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut8] <- "H"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut7] <- "AE"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut7] <- "AF"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut7] <- "AG"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut7] <- "AH"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut7] <- "AI"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut7] <- "AJ"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut7] <- "AK"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut7] <- "AA"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut7] <- "Q"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut7] <- "G"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut6] <- "AO"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut6] <- "AP"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut6] <- "AQ"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut6] <- "AR"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut6] <- "AS"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut6] <- "AT"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut6] <- "AJ"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut6] <- "Z"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut6] <- "P"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut6] <- "F"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut5] <- "AY"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut5] <- "AZ"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut5] <- "BA"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut5] <- "BB"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut5] <- "BC"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut5] <- "AS"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut5] <- "AI"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut5] <- "Y"
```

```
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut5] <- "O"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut5] <- "E"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut4] <- "BI"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut4] <- "BJ"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut4] <- "BK"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut4] <- "BL"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut4] <- "BB"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut4] <- "AR"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut4] <- "AH"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut4] <- "X"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut4] <- "N"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut4] <- "D"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut3] <- "BS"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut3] <- "BT"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut3] <- "BU"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut3] <- "BK"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut3] <- "BA"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut3] <- "AQ"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut3] <- "AG"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut3] <- "W"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut3] <- "M"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut3] <- "C"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut2] <- "CC"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut2] <- "CD"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut2] <- "BT"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut2] <- "BJ"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut2] <- "AZ"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut2] <- "AP"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut2] <- "AF"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut2] <- "V"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut2] <- "L"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut2] <- "B"


data_TAP$aggression[data_TAP$A == data_TAP$cut1 & data_TAP$B == data_TAP$cut1] <- "CM"
data_TAP$aggression[data_TAP$A == data_TAP$cut2 & data_TAP$B == data_TAP$cut1] <- "CC"
data_TAP$aggression[data_TAP$A == data_TAP$cut3 & data_TAP$B == data_TAP$cut1] <- "BS"
data_TAP$aggression[data_TAP$A == data_TAP$cut4 & data_TAP$B == data_TAP$cut1] <- "BI"
data_TAP$aggression[data_TAP$A == data_TAP$cut5 & data_TAP$B == data_TAP$cut1] <- "AY"
data_TAP$aggression[data_TAP$A == data_TAP$cut6 & data_TAP$B == data_TAP$cut1] <- "AO"
data_TAP$aggression[data_TAP$A == data_TAP$cut7 & data_TAP$B == data_TAP$cut1] <- "AE"
data_TAP$aggression[data_TAP$A == data_TAP$cut8 & data_TAP$B == data_TAP$cut1] <- "U"
data_TAP$aggression[data_TAP$A == data_TAP$cut9 & data_TAP$B == data_TAP$cut1] <- "K"
data_TAP$aggression[data_TAP$A == data_TAP$cut10 & data_TAP$B == data_TAP$cut1] <- "A"

head(data_TAP)
```

```
##   id time A  B cut1 cut2 cut3 cut4 cut5 cut6 cut7 cut8 cut9 cut10 aggression
## 1  1    1 0  0    0   10   20   30   40   50   60   70   80    90         CM
## 2  1    2 0 10    0   10   20   30   40   50   60   70   80    90         CC
## 3  1    3 0  0    0   10   20   30   40   50   60   70   80    90         CM
## 4  1    4 0  0    0   10   20   30   40   50   60   70   80    90         CM
## 5  1    5 0  0    0   10   20   30   40   50   60   70   80    90         CM
## 6  1    6 0  0    0   10   20   30   40   50   60   70   80    90         CM
```

We made adjustments to accommodate our 10 x 10 grid for indistinguishable dyads and to suit our layout preferences. The PDF file containing all dyads' visual representations are included in the repository.

```
all_ids <- unique(ind_TAP$id)

# Remove NAs from all_ids

all_ids <- na.omit(all_ids)

cut1 <- -5
cut2 <- 5
cut3 <- 15
cut4 <- 25
cut5 <- 35
cut6 <- 45
cut7 <- 55
cut8 <- 65
cut9 <- 75
cut10 <- 85
cut10 <- 95


# open the pdf file
pdf("GSA_all_ind.pdf", width = 10, height = 7)

for(i in 1:length(all_ids)){
  dyad_id <- all_ids[i]
  data_loop_subset <- subset(ind_TAP, ind_TAP$id == dyad_id)

#let's remove the missing data

data_loop_noNA <- na.omit(data_loop_subset)

#setwd("~/Desktop")

# At this point we need to check if there is actually data for the individual and only
#plot points where there are data for both members of the dyad at each time point.

if(nrow(data_loop_noNA) == 0){}

if(nrow(data_loop_noNA) > 0){

plot_title = paste('Dyad ID = ', unique(data_loop_noNA$id), sep = '') #adding a dyad ID
#label at the top of each plot
grid_plot <-
```

```
ggplot(data = data_loop_noNA, aes(x = x, y = y)) +
  xlim(-5,95) + #giving a little extra room (the scale is 0-100) so points of the edge
  #of the scale will show up easily
  ylim(-5,95) + #giving a little extra room (the scale is 0-100) so points of the edge
  #of the scale will show up easily
  ylab('Aggression Sibling 2') + #y-axis label
  xlab('Aggression Sibling 1') + #x-axis label
  geom_rect(xmin=-5,xmax=5,ymin=-5,ymax=5, fill="#f95f32") +
  geom_rect(xmin=-5,xmax=5,ymin=5,ymax=15, fill="#f95740") +
  geom_rect(xmin=-5,xmax=5,ymin=15,ymax=25, fill="#f75b53") +
  geom_rect(xmin=-5,xmax=5,ymin=25,ymax=35, fill="#f56264") +
  geom_rect(xmin=-5,xmax=5,ymin=35,ymax=45, fill="#e06586") +
  geom_rect(xmin=-5,xmax=5,ymin=45,ymax=55, fill="#cf6596") +
  geom_rect(xmin=-5,xmax=5,ymin=55,ymax=65, fill="#b76b9d") +
  geom_rect(xmin=-5,xmax=5,ymin=65,ymax=75, fill="#9f689d") +
  geom_rect(xmin=-5,xmax=5,ymin=75,ymax=85, fill="#927bab") +
  geom_rect(xmin=-5,xmax=5,ymin=85,ymax=95, fill="#8a83ac") +

  geom_rect(xmin=5,xmax=15,ymin=-5,ymax=5, fill="#f95740") +
  geom_rect(xmin=5,xmax=15,ymin=5,ymax=15, fill="#f94d34") +
  geom_rect(xmin=5,xmax=15,ymin=15,ymax=25, fill="#f74a41") +
  geom_rect(xmin=5,xmax=15,ymin=25,ymax=35, fill="#ef5a62") +
  geom_rect(xmin=5,xmax=15,ymin=35,ymax=45, fill="#de587d") +
  geom_rect(xmin=5,xmax=15,ymin=45,ymax=55, fill="#cd5a8f") +
  geom_rect(xmin=5,xmax=15,ymin=55,ymax=65, fill="#b7629b") +
  geom_rect(xmin=5,xmax=15,ymin=65,ymax=75, fill="#9f619c") +
  geom_rect(xmin=5,xmax=15,ymin=75,ymax=85, fill="#8e6ca2") +
  geom_rect(xmin=5,xmax=15,ymin=85,ymax=95, fill="#8177ac") +

  geom_rect(xmin=15,xmax=25,ymin=-5,ymax=5, fill="#f75b53") +
  geom_rect(xmin=15,xmax=25,ymin=5,ymax=15, fill="#f74a41") +
  geom_rect(xmin=15,xmax=25,ymin=15,ymax=25, fill="#f74339") +
  geom_rect(xmin=15,xmax=25,ymin=25,ymax=35, fill="#ee4d56") +
  geom_rect(xmin=15,xmax=25,ymin=35,ymax=45, fill="#e04e76") +
  geom_rect(xmin=15,xmax=25,ymin=45,ymax=55, fill="#cf4f8a") +
  geom_rect(xmin=15,xmax=25,ymin=55,ymax=65, fill="#b75b99") +
  geom_rect(xmin=15,xmax=25,ymin=65,ymax=75, fill="#9f5a9c") +
  geom_rect(xmin=15,xmax=25,ymin=75,ymax=85, fill="#8b649d") +
  geom_rect(xmin=15,xmax=25,ymin=85,ymax=95, fill="#786cab") +

  geom_rect(xmin=25,xmax=35,ymin=-5,ymax=5, fill="#f56264") +
  geom_rect(xmin=25,xmax=35,ymin=5,ymax=15, fill="#ef5a62") +
  geom_rect(xmin=25,xmax=35,ymin=15,ymax=25, fill="#ee4d56") +
  geom_rect(xmin=25,xmax=35,ymin=25,ymax=35, fill="#ef3f49") +
  geom_rect(xmin=25,xmax=35,ymin=35,ymax=45, fill="#e0446f") +
  geom_rect(xmin=25,xmax=35,ymin=45,ymax=55, fill="#cd4685") +
  geom_rect(xmin=25,xmax=35,ymin=55,ymax=65, fill="#b94f97") +
  geom_rect(xmin=25,xmax=35,ymin=65,ymax=75, fill="#a1539e") +
  geom_rect(xmin=25,xmax=35,ymin=75,ymax=85, fill="#87599c") +
  geom_rect(xmin=25,xmax=35,ymin=85,ymax=95, fill="#7161ab") +

  geom_rect(xmin=35,xmax=45,ymin=-5,ymax=5, fill="#e06586") +
  geom_rect(xmin=35,xmax=45,ymin=5,ymax=15, fill="#de587d") +
```

```
geom_rect(xmin=35,xmax=45,ymin=15,ymax=25, fill="#e04e76") +
geom_rect(xmin=35,xmax=45,ymin=25,ymax=35, fill="#e0446f") +
geom_rect(xmin=35,xmax=45,ymin=35,ymax=45, fill="#df3866") +
geom_rect(xmin=35,xmax=45,ymin=45,ymax=55, fill="#cd3e81") +
geom_rect(xmin=35,xmax=45,ymin=55,ymax=65, fill="#b74994") +
geom_rect(xmin=35,xmax=45,ymin=65,ymax=75, fill="#a14c9e") +
geom_rect(xmin=35,xmax=45,ymin=75,ymax=85, fill="#84509c") +
geom_rect(xmin=35,xmax=45,ymin=85,ymax=95, fill="#6b58ac") +

geom_rect(xmin=45,xmax=55,ymin=-5,ymax=5, fill="#cf6596") +
geom_rect(xmin=45,xmax=55,ymin=5,ymax=15, fill="#cd5a8f") +
geom_rect(xmin=45,xmax=55,ymin=15,ymax=25, fill="#cf4f8a") +
geom_rect(xmin=45,xmax=55,ymin=25,ymax=35, fill="#cd4685") +
geom_rect(xmin=45,xmax=55,ymin=35,ymax=45, fill="#cd3e81") +
geom_rect(xmin=45,xmax=55,ymin=45,ymax=55, fill="#cd347b") +
geom_rect(xmin=45,xmax=55,ymin=55,ymax=65, fill="#b63e91") +
geom_rect(xmin=45,xmax=55,ymin=65,ymax=75, fill="#a1419d") +
geom_rect(xmin=45,xmax=55,ymin=75,ymax=85, fill="#83489d") +
geom_rect(xmin=45,xmax=55,ymin=85,ymax=95, fill="#644fac") +


geom_rect(xmin=55,xmax=65,ymin=-5,ymax=5, fill="#b76b9d") +
geom_rect(xmin=55,xmax=65,ymin=5,ymax=15, fill="#b7629b") +
geom_rect(xmin=55,xmax=65,ymin=15,ymax=25, fill="#b75b99") +
geom_rect(xmin=55,xmax=65,ymin=25,ymax=35, fill="#b94f97") +
geom_rect(xmin=55,xmax=65,ymin=35,ymax=45, fill="#b74994") +
geom_rect(xmin=55,xmax=65,ymin=45,ymax=55, fill="#b63e91") +
geom_rect(xmin=55,xmax=65,ymin=55,ymax=65, fill="#b2308a") +
geom_rect(xmin=55,xmax=65,ymin=65,ymax=75, fill="#a0359b") +
geom_rect(xmin=55,xmax=65,ymin=75,ymax=85, fill="#7f409c") +
geom_rect(xmin=55,xmax=65,ymin=85,ymax=95, fill="#5a42ab") +


geom_rect(xmin=65,xmax=75,ymin=-5,ymax=5, fill="#9f689d") +
geom_rect(xmin=65,xmax=75,ymin=5,ymax=15, fill="#9f619c") +
geom_rect(xmin=65,xmax=75,ymin=15,ymax=25, fill="#9f5a9c") +
geom_rect(xmin=65,xmax=75,ymin=25,ymax=35, fill="#a1539e") +
geom_rect(xmin=65,xmax=75,ymin=35,ymax=45, fill="#a14c9e") +
geom_rect(xmin=65,xmax=75,ymin=45,ymax=55, fill="#a1419d") +
geom_rect(xmin=65,xmax=75,ymin=55,ymax=65, fill="#a0359b") +
geom_rect(xmin=65,xmax=75,ymin=65,ymax=75, fill="#a01d9a") +
geom_rect(xmin=65,xmax=75,ymin=75,ymax=85, fill="#7c319d") +
geom_rect(xmin=65,xmax=75,ymin=85,ymax=95, fill="#5439ac") +


geom_rect(xmin=75,xmax=85,ymin=-5,ymax=5, fill="#927bab") +
geom_rect(xmin=75,xmax=85,ymin=5,ymax=15, fill="#8e6ca2") +
geom_rect(xmin=75,xmax=85,ymin=15,ymax=25, fill="#8b649d") +
geom_rect(xmin=75,xmax=85,ymin=25,ymax=35, fill="#87599c") +
geom_rect(xmin=75,xmax=85,ymin=35,ymax=45, fill="#84509c") +
geom_rect(xmin=75,xmax=85,ymin=45,ymax=55, fill="#83489d") +
geom_rect(xmin=75,xmax=85,ymin=55,ymax=65, fill="#7f409c") +
geom_rect(xmin=75,xmax=85,ymin=65,ymax=75, fill="#7c319d") +
```

```
  geom_rect(xmin=75,xmax=85,ymin=75,ymax=85, fill="#72169b") +
  geom_rect(xmin=75,xmax=85,ymin=85,ymax=95, fill="#4c2fab") +


  geom_rect(xmin=85,xmax=95,ymin=-5,ymax=5, fill="#8a83ac") +
  geom_rect(xmin=85,xmax=95,ymin=5,ymax=15, fill="#8177ac") +
  geom_rect(xmin=85,xmax=95,ymin=15,ymax=25, fill="#786cab") +
  geom_rect(xmin=85,xmax=95,ymin=25,ymax=35, fill="#7161ab") +
  geom_rect(xmin=85,xmax=95,ymin=35,ymax=45, fill="#6b58ac") +
  geom_rect(xmin=85,xmax=95,ymin=45,ymax=55, fill="#644fac") +
  geom_rect(xmin=85,xmax=95,ymin=55,ymax=65, fill="#5a42ab") +
  geom_rect(xmin=85,xmax=95,ymin=65,ymax=75, fill="#5439ac") +
  geom_rect(xmin=85,xmax=95,ymin=75,ymax=85, fill="#4c2fab") +
  geom_rect(xmin=85,xmax=95,ymin=85,ymax=95, fill="#4322ad") +

   theme(
    axis.text = element_text(size = 14, color = 'black'),
    axis.title = element_text(size = 18),
    panel.grid.major = element_line(colour = "white"),
    panel.grid.minor = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank()
) + #adjusting size of text and titles in plot
  geom_vline(xintercept = c(-5,5,15,25,35,45,55,65,75,85,95)) + #adding vertical lines
  #to distinguish cells of grid
  geom_hline(yintercept = c(-5,5,15,25,35,45,55,65,75,85,95)) + #adding horizontal lines
  #to distinguish cells of grid
    geom_point(colour= "black") + #setting points on plot to white
  geom_path(colour = "black")  + #setting path connecting points to the color white
scale_x_continuous(breaks = scales::pretty_breaks(n = 10),limits=c(0,90)) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10),limits=c(0,90)) +
  ggtitle(plot_title) #adding title to plot
print(grid_plot)
}
}

dev.off()
```

2. Creating Sequences. We renamed variable names and column names to align with our specific dataset.

```
#subsetting out the data we need
data_sub <- data_TAP[ ,c("id", "time", "aggression")]

#Reformatting data: long to wide.
  data_wide <- reshape(data=data_sub,
                       timevar=c("time"),            #time variable
                       idvar= c("id"),               #id variable
                       v.names=c("aggression"),      #repeated measures variable
                       direction="wide", sep=".")

head(data_wide)
```

```
##     id aggression.1 aggression.2 aggression.3 aggression.4 aggression.5
```

```
## 1    1           CM           CC           CM           CM           CM
## 61   2           CC           CC           CM           BS           CC
## 121  3           CM           CM           CM           CM           CM
## 181  4           CM           BT           CD           AQ           AO
## 241  5           CM            J            J            A            A
## 301  6           CM           AY           AQ           AG           AO
##     aggression.6 aggression.7 aggression.8 aggression.9 aggression.10
## 1             CM           CM           CM           CM            CM
## 61            CM           CC           CC           BS             A
## 121           CM           CM           CM           CM            CM
## 181           AO           BI           BK           AQ            AY
## 241            J            J            J            J             J
## 301           BT           AP           AP           BB            AO
##     aggression.11 aggression.12 aggression.13 aggression.14 aggression.15
## 1              CM            CM            CM            CM            CM
## 61              A             B             A            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            AY            CC            BS            AO            AP
## 241             J             J             A             J             J
## 301            BB            BB            AZ            AZ            CD
##     aggression.16 aggression.17 aggression.18 aggression.19 aggression.20
## 1              CM            CM            CM            CM            CM
## 61             CC            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            AP            BI            BI             N             N
## 241             J             J             J             J             J
## 301            BS            AZ            AP            AQ            AP
##     aggression.21 aggression.22 aggression.23 aggression.24 aggression.25
## 1              CM            CM            CM            CM            CM
## 61             CM            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            AY            BI            BK            BS            BI
## 241             J             J             J             J             J
## 301            CD            CD            CM            CC            CD
##     aggression.26 aggression.27 aggression.28 aggression.29 aggression.30
## 1              CM            CM            CM            CM            CM
## 61             CM            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            BI            BI            BS            BK            BK
## 241             J             J             J             J             J
## 301            BJ            AZ            AZ            BK            BL
##     aggression.31 aggression.32 aggression.33 aggression.34 aggression.35
## 1               A            AO            AY            CM            CM
## 61              D             E             A             A             B
## 121            CM            CM            CM            CM            CM
## 181             G             G             J             A             A
## 241             J             J             J             J             J
## 301            AJ            AJ             F            AQ            AQ
##     aggression.36 aggression.37 aggression.38 aggression.39 aggression.40
## 1              CM            CM            CM            CM            CM
## 61              D             B             A            CD            CD
## 121            CM            CM            CM            CM            CM
## 181             D             A            AO            AS             C
## 241             J             J             J             J             J
```

```
## 301            AQ             F            AR             P            AJ
##     aggression.41 aggression.42 aggression.43 aggression.44 aggression.45
## 1              CM            CM            CM            CM            CM
## 61             CD            CD            CD            CC            CC
## 121            CM            CM            CM            CM            CM
## 181             H             I             G             Q            AK
## 241             J             J             J             J             J
## 301             F             F            AJ             P            AJ
##     aggression.46 aggression.47 aggression.48 aggression.49 aggression.50
## 1              CM            CM            CM            CM            CM
## 61             CM            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            AY            BC            AS             K             U
## 241             J             J             J             J             J
## 301            AQ            AQ            AT             P             Z
##     aggression.51 aggression.52 aggression.53 aggression.54 aggression.55
## 1              CM            CM            CM            CM            CM
## 61             CM            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181            AA            AK            AJ            AH             E
## 241             J             J             J             J             J
## 301            AG             F            AR            AI            AA
##     aggression.56 aggression.57 aggression.58 aggression.59 aggression.60
## 1              CM            CM            CM            CM            CM
## 61             CM            CM            CM            CM            CM
## 121            CM            CM            CM            CM            CM
## 181             F             I             G             P             Z
## 241             J             J             J             J             J
## 301             Z            AJ            AT            AJ            AT
```

We inserted alphabet and color labels corresponding to the 10 x 10 cell grid, encompassing the full 100 grids, which enables us to utilize it for both indistinguishable and distinguishable analysis.

```
# Creating alphabet.

#this object contains the letters that appear in the data set.
gs.alphabet <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",
                 "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "AA", "AB",
                 "AC", "AD", "AE", "AF", "AG", "AH", "AI", "AJ", "AK", "AL", "AM", "AN",
                 "AO", "AP", "AQ", "AR", "AS", "AT", "AU", "AV", "AW", "AX", "AY", "AZ",
                 "BA", "BB", "BC", "BD", "BE", "BF", "BG", "BH", "BI", "BJ", "BK", "BL",
                 "BM", "BN", "BO", "BP", "BQ", "BR", "BS", "BT", "BU", "BV", "BW", "BX",
                 "BY", "BZ", "CA", "CB", "CC", "CD", "CE", "CF", "CG", "CH", "CI", "CJ",
                 "CK", "CL", "CM", "CN", "CO", "CP", "CQ", "CR", "CS", "CT", "CU", "CV")

#this object allows for more helpful labels if applicable (e.g., negative, neutral, and
#positive affect).
gs.labels <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O",
               "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "AA", "AB", "AC",
               "AD", "AE", "AF", "AG", "AH", "AI", "AJ", "AK", "AL", "AM", "AN", "AO",
               "AP", "AQ", "AR", "AS", "AT", "AU", "AV", "AW", "AX", "AY", "AZ", "BA",
               "BB", "BC", "BD", "BE", "BF", "BG", "BH", "BI", "BJ", "BK", "BL", "BM",
               "BN", "BO", "BP", "BQ", "BR", "BS", "BT", "BU", "BV", "BW", "BX", "BY",
               "BZ", "CA", "CB", "CC", "CD", "CE", "CF", "CG", "CH", "CI", "CJ", "CK",
```

```
                "CL", "CM", "CN", "CO", "CP", "CQ", "CR", "CS", "CT", "CU", "CV")



#Creating the sequences.

A <- "#8a83ac"
B <- "#8177ac"
C <- "#786cab"
D <- "#7161ab"
E <- "#6b58ac"
F <- "#644fac"
G <- "#5a42ab"
H <- "#5439ac"
I <- "#4c2fab"
J <- "#4322ad"

K <- "#927bab"
L <- "#8e6ca2"
M <- "#8b649d"
N <- "#87599c"
O <- "#84509c"
P <- "#83489d"
Q <- "#7f409c"
R <- "#7c319d"
S <- "#72169b"
T <- "#4c2fab"

U <- "#9f689d"
V <- "#9f619c"
W <- "#9f5a9c"
X <- "#a1539e"
Y <- "#a14c9e"
Z <- "#a1419d"
AA <- "#a0359b"
AB <- "#a01d9a"
AC <- "#7c319d"
AD <- "#5439ac"

AE <- "#b76b9d"
AF <- "#b7629b"
AG <- "#b75b99"
AH <- "#b94f97"
AI <- "#b74994"
AJ <- "#b63e91"
AK <- "#b2308a"
AL <- "#a0359b"
AM <- "#7f409c"
AN <- "#5a42ab"

AO <- "#cf6596"
AP <- "#cd5a8f"
AQ <- "#cf4f8a"
```

```
AR <- "#cd4685"
AS <- "#cd3e81"
AT <- "#cd347b"
AU <- "#b63e91"
AV <- "#a1419d"
AW <- "#83489d"
AX <- "#644fac"

AY <- "#e06586"
AZ <- "#de587d"
BA <- "#e04e76"
BB <- "#e0446f"
BC <- "#df3866"
BD <- "#cd3e81"
BE <- "#b74994"
BF <- "#a14c9e"
BG <- "#84509c"
BH <- "#6b58ac"

BI <- "#f56264"
BJ <- "#ef5a62"
BK <- "#ee4d56"
BL <- "#ef3f49"
BM <- "#e0446f"
BN <- "#cd4685"
BO <- "#b94f97"
BP <- "#a1539e"
BQ <- "#87599c"
BR <- "#7161ab"

BS <- "#f75b53"
BT <- "#f74a41"
BU <- "#f74339"
BV <- "#ee4d56"
BW <- "#e04e76"
BX <- "#cf4f8a"
BY <- "#b75b99"
BZ <- "#9f5a9c"
CA <- "#8b649d"
CB <- "#786cab"

CC <- "#f95740"
CD <- "#f94d34"
CE <- "#f74a41"
CF <- "#ef5a62"
CG <- "#de587d"
CH <- "#cd5a8f"
CI <- "#b7629b"
CJ <- "#9f619c"
CK <- "#8e6ca2"
CL <- "#8177ac"

CM <- "#f95f32"
```

```
CN <- "#f95740"
CO <- "#f75b53"
CP <- "#f56264"
CQ <- "#e06586"
CR <- "#cf6596"
CS <- "#b76b9d"
CT <- "#9f689d"
CU <- "#927bab"
CV <- "#8a83ac"

#These are assigning colors to each letter. seqdef (the function below) also has default
#colors.
```

We replaced the original alphabet labels with our custom alphabet labels and made necessary adjustments to the columns specifying the repeated measures data.

```
#this creates an object that contains all of the sequences.

#seqdef is a function in TraMineR
#input: data, columns containing repeated measures data, alphabet, labels, xtstep = steps
#between tick marks, cpal (colors)
aggression.seq <- seqdef(data_wide, var = 2:60, alphabet = gs.alphabet,
                    labels = gs.labels, xtstep = 6, cpal=c(A, B, C, D, E, F, G, H, I, J, K,
                                                           L, M, N, O, P, Q, R, S, T, U, V,
                                                           W, X, Y, Z, AA, AB, AC, AD, AE,
                                                           AF, AG, AH, AI, AJ, AK, AL, AM,
                                                           AN, AO, AP, AQ, AR, AS, AT, AU,
                                                           AV, AW, AX, AY, AZ, BA, BB, BC,
                                                           BD, BE, BF, BG, BH, BI, BJ, BK,
                                                           BL, BM, BN, BO, BP, BQ, BR, BS,
                                                           BT, BU, BV, BW, BX, BY, BZ, CA,
                                                           CB, CC, CD, CE, CF, CG, CH, CI,
                                                           CJ, CK, CL, CM, CN, CO, CP, CQ,
                                                           CR, CS, CT, CU, CV))
```

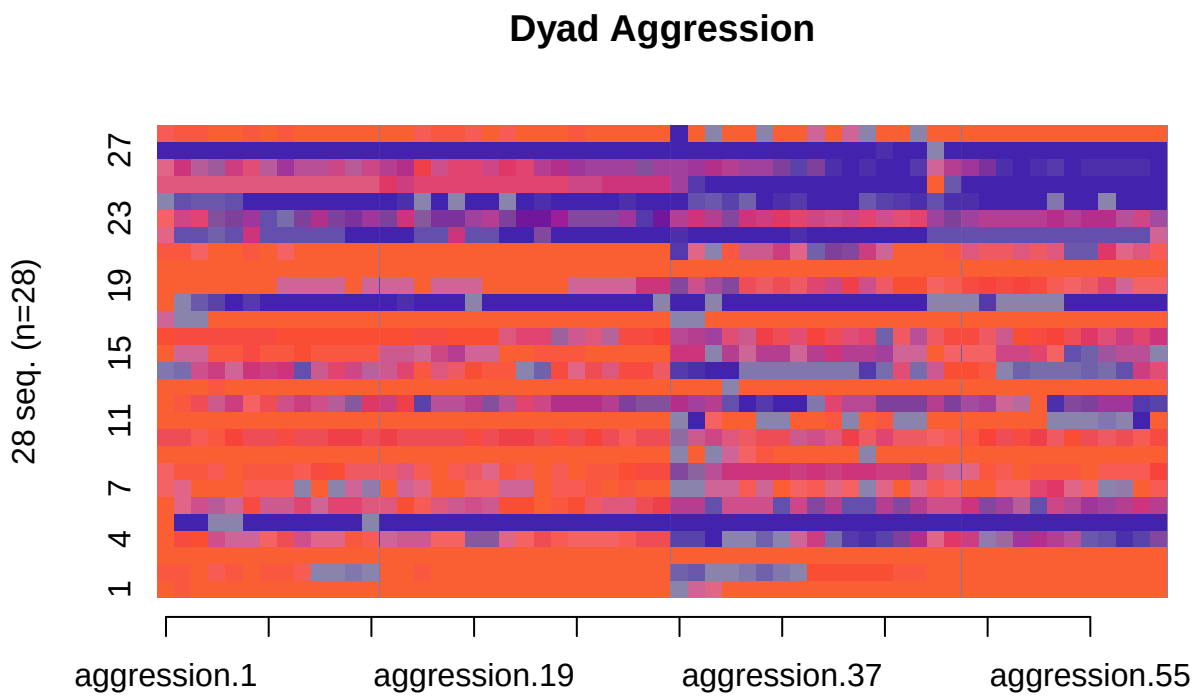
```
##  [>] 55 distinct states appear in the data:

##      1 = A

##      2 = AA

##      3 = AB

##      4 = AE

##      5 = AF

##      6 = AG

##      7 = AH
```

```
##      8 = AI
##      9 = AJ
##      10 = AK
##      11 = AO
##      12 = AP
##       ...
##  [>] state coding:
##        [alphabet]  [label]  [long label]
##      1  A           A        A
##      2  B           B        B
##      3  C           C        C
##      4  D           D        D
##      5  E           E        E
##      6  F           F        F
##      7  G           G        G
##      8  H           H        H
##      9  I           I        I
##      10  J           J        J
##      11  K           K        K
##      12  L           L        L
##       ... (100 states)
##  [>] 28 sequences in the data set
##  [>] min/max sequence length: 59/59
```

```
#Plotting the sequences.

#seqIplot is a function in TraMineR
#input: sequence object (created above), legend, title

seqIplot(aggression.seq, withlegend = FALSE, title="Dyad Aggression")
```

```
##  [!!] In rmarkdown::render() : title is deprecated, use main instead.
```

```
##  [!!] In rmarkdown::render() : withlegend is deprecated, use with.legend instead.
```




```
#Setting up cost matrix.

#example: constant cost matrix

#we are not actually using this cost matrix, but it creates some nice labels that we will
#use later
costmatrix <- seqsubm(aggression.seq, method="CONSTANT", cval=2, with.missing=TRUE,
        miss.cost=.5, time.varying=FALSE, weighted=TRUE,
        transition="both", lag=1)
```

```
##  [!!] seqcost: 'with.missing' set as FALSE as 'seqdata' has no non-void missing values
```

```
##  [>] creating 100x100 substitution-cost matrix using 2 as constant value
```

```
costmatrix
```

```
##    A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG AH
## A  0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## B  2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## C  2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## D  2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## E  2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## F  2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## G  2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## H  2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## I  2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## J  2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## K  2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## L  2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## M  2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## N  2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## O  2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## P  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## Q  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## R  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## S  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## T  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## U  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2  2  2  2  2  2  2  2  2
## V  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2  2  2  2  2  2  2  2  2
## W  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2  2  2  2  2  2  2  2  2
## X  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2  2  2  2  2  2  2  2  2
## Y  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2  2  2  2  2  2  2  2  2
## Z  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0  2  2  2  2  2  2  2  2
## AA 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  0  2  2  2  2  2  2  2
## AB 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  0  2  2  2  2  2  2
## AC 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  0  2  2  2  2  2
## AD 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  0  2  2  2  2
## AE 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  0  2  2  2
## AF 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  0  2  2
## AG 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  0  2
## AH 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  0
## AI 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AJ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AK 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AM 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AN 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AO 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AP 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AQ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AR 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AS 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AT 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AU 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AV 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AW 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AX 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## AY 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
```

```
## AZ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BA 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BB 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BC 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BD 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BE 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BF 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BG 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BH 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BI 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BJ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BK 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BM 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BN 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BO 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BP 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BQ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BR 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BS 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BT 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BU 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BV 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BW 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BX 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BY 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## BZ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CA 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CB 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CC 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CD 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CE 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CF 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CG 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CH 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CI 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CJ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CK 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CM 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CN 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CO 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CP 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CQ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CR 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CS 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CT 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CU 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
## CV 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2
##    AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX AY AZ BA BB BC BD BE BF BG
## A   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## B   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## C   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## D   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

```
## E  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## F  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## G  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## H  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## I  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## J  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## K  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## L  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## M  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## N  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## O  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## P  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Q  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## R  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## S  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## T  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## U  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## V  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## W  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## X  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Y  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Z  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AA 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AB 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AC 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AD 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AE 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AF 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AG 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AH 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AI 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AJ 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AK 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AL 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AM 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AN 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AO 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AP 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AQ 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AR 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AS 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AT 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## AU 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2
## AV 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2
## AW 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2
## AX 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2
## AY 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2
## AZ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2
## BA 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2
## BB 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2
## BC 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2
## BD 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2
## BE 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2
## BF 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2
```

```
## BG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0
## BH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BW  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BX  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BY  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BZ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##     BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD CE CF
## A    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## B    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## C    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## D    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## E    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## F    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## G    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## H    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## I    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## J    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## K    2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

```
## L   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## M   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## N   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## O   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## P   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Q   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## R   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## S   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## T   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## U   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## V   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## W   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## X   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Y   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Z   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AW  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AX  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AY  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AZ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BH  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BI  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BJ  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BK  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BL  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BM  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

```
## BN  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BO  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BP  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BQ  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BR  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2
## BS  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2
## BT  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2
## BU  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2
## BV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2
## BW  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2
## BX  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2
## BY  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2
## BZ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2
## CA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2
## CB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2
## CC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2
## CD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2
## CE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0
## CF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##    CG CH CI CJ CK CL CM CN CO CP CQ CR CS CT CU CV
## A   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## B   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## C   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## D   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## E   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## F   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## G   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## H   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## I   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## J   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## K   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## L   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## M   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## N   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## O   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## P   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Q   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## R   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

```
## S   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## T   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## U   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## V   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## W   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## X   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Y   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## Z   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AW  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AX  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AY  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## AZ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BG  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BH  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BI  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BJ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BK  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BL  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BM  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BN  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BO  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BP  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BQ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BR  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BS  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BT  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

```
## BU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BW  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BX  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BY  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## BZ  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CA  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CB  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CC  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CD  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CE  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CF  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CG  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CH  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## CI  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2  2
## CJ  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2  2
## CK  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2  2
## CL  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2  2
## CM  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2  2
## CN  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2  2
## CO  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2  2
## CP  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2  2
## CQ  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2  2
## CR  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2  2
## CS  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2  2
## CT  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2  2
## CU  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0  2
## CV  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  0
```

We calculated a cost matrix tailored for our 10 x 10 grid and dataset in Matlab using by performing matrix multiplication of cell rows and columns.

```
#defining cost matrix, it will be (n+1) by (n+1) with n = number of cells in the grid
costmatrix2 <- matrix(c(0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,
                        7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,
                        4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,
                        10.0,11.0,12.0,13.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,
                        6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,7.0,8.0,9.0,10.0,
                        11.0,12.0,13.0,14.0,15.0,16.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,
                        15.0,16.0,17.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0,
1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,7.0,6.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,14.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,9.0,8.0,
9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,10.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,
2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,7.0,6.0,7.0,8.0,
9.0,10.0,11.0,12.0,13.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,10.0,9.0,8.0,9.0,
10.0,11.0,12.0,13.0,14.0,15.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,11.0,10.0,9.0,8.0,9.0,10.0,
```

11.0,12.0,13.0,14.0,12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,4.0,3.0,2.0,1.0,0.0,
1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,
5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,
9.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,
11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,
13.0,13.0,12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,14.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,
3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,
8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,10.0,9.0,
8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,
9.0,8.0,7.0,8.0,9.0,10.0,11.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,14.0,13.0,
12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,
6.0,5.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,13.0,12.0,11.0,10.0,9.0,
8.0,7.0,8.0,9.0,10.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,10.0,11.0,12.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,
3.0,2.0,1.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
3.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,
6.0,7.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,
8.0,9.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,
9.0,10.0,11.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,
2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,
14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,
8.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,17.0,16.0,15.0,14.0,13.0,12.0,11.0,
10.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,
5.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,
8.0,7.0,17.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,18.0,17.0,16.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,7.0,8.0,
9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,8.0,7.0,8.0,9.0,10.0,11.0,
12.0,13.0,14.0,15.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,3.0,2.0,1.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,
7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,10.0,
9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,11.0,10.0,9.0,8.0,9.0,10.0,
11.0,12.0,13.0,14.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,
3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,
7.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,9.0,
8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,11.0,10.0,

9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,6.0,5.0,
4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,
2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,
4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,
8.0,9.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,
11.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,
7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,10.0,9.0,8.0,
7.0,6.0,5.0,4.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,7.0,8.0,9.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,14.0,13.0,12.0,11.0,
10.0,9.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,
2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,
7.0,8.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,
8.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,
1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,15.0,14.0,
13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,8.0,
7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,14.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,16.0,15.0,14.0,
13.0,12.0,11.0,10.0,9.0,8.0,7.0,17.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,7.0,8.0,9.0,10.0,11.0,
12.0,13.0,14.0,15.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,3.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,8.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,10.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,
12.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,
14.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,
2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,7.0,8.0,
9.0,10.0,11.0,12.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,6.0,5.0,4.0,3.0,2.0,3.0,
4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,
4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,
7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,
7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,11.0,10.0,9.0,
8.0,7.0,8.0,9.0,10.0,11.0,12.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,9.0,
8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,
8.0,7.0,6.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,5.0,

4.0,3.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,
1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,
3.0,2.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,
2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,13.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,5.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,7.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,11.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
1.0,0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,14.0,
13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,16.0,
15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,9.0,10.0,11.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,
13.0,14.0,15.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,5.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,
3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,7.0,6.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,
4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,8.0,
7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,7.0,6.0,
5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,
3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,
10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,
3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,
7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,
6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,
4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,
4.0,3.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,
5.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,
3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,9.0,

8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,7.0,8.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
5.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,
7.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,12.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,9.0,10.0,11.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,7.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,0.0,1.0,
2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,
6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,
6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,
3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,
3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,
7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,
3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,
3.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,
6.0,7.0,8.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,9.0,8.0,
7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,12.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,

5.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,
4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,12.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,9.0,10.0,11.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,7.0,6.0,5.0,
6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,0.0,1.0,
2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,
6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,
10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,
6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,
2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,
2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,
6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,
9.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,
6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,
2.0,1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,
2.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,
6.0,7.0,8.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,
6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,
7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,12.0,11.0,10.0,9.0,8.0,
7.0,6.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,
4.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,
2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,5.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,13.0,12.0,11.0,10.0,9.0,8.0,
7.0,6.0,5.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,
4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,
0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,
6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,

14.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,
12.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,7.0,6.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,5.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,10.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,
13.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,
5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,
9.0,10.0,11.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,
8.0,9.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,
3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,
3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,
6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,4.0,
5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,
5.0,6.0,7.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,
5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,7.0,
6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,10.0,9.0,
8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,
4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,1.0,
2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,
7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,
6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,
7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,4.0,5.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,13.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,6.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,4.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,15.0,
14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,
13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,14.0,15.0,16.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,8.0,7.0,8.0,9.0,10.0,11.0,
12.0,13.0,14.0,15.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,5.0,6.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,3.0,4.0,5.0,6.0,7.0,

8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
9.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,
8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,
6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,4.0,
3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,0.0,
1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,9.0,8.0,7.0,6.0,7.0,8.0,
9.0,10.0,11.0,12.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,
8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,
3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,11.0,10.0,9.0,
8.0,7.0,8.0,9.0,10.0,11.0,12.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,
5.0,6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,3.0,4.0,
5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,
5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,
6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,11.0,
10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,
7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,
3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,
1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,
5.0,6.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,
8.0,9.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,
7.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,
6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,
4.0,3.0,2.0,1.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,14.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,8.0,9.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,6.0,7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,4.0,5.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,
3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,
14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,3.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,11.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,3.0,2.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,14.0,15.0,16.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,5.0,6.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,14.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,8.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,
3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,1.0,
2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,2.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,9.0,8.0,7.0,8.0,
9.0,10.0,11.0,12.0,13.0,14.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,7.0,6.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,3.0,4.0,5.0,

6.0,7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,
6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,
8.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,
12.0,13.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,
11.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,5.0,
4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,1.0,
0.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,12.0,11.0,10.0,9.0,
8.0,9.0,10.0,11.0,12.0,13.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,10.0,9.0,8.0,7.0,
6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,4.0,
5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,
5.0,6.0,7.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,
5.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,
12.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,
10.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,8.0,
7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,4.0,
3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,6.0,5.0,4.0,3.0,2.0,
1.0,2.0,3.0,4.0,5.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,13.0,12.0,11.0,10.0,
9.0,8.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,8.0,7.0,
6.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,
4.0,6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,15.0,
14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,
13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,
11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,7.0,6.0,5.0,
4.0,3.0,2.0,1.0,0.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,16.0,15.0,14.0,13.0,
12.0,11.0,10.0,9.0,8.0,9.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,7.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,12.0,11.0,10.0,
9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,
2.0,1.0,0.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,17.0,16.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,7.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,1.0,0.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,10.0,11.0,12.0,13.0,
14.0,15.0,16.0,17.0,18.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,14.0,15.0,16.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,5.0,6.0,7.0,
8.0,9.0,10.0,11.0,12.0,13.0,14.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,3.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,11.0,12.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,1.0,2.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,9.0,10.0,11.0,12.0,
13.0,14.0,15.0,16.0,17.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,8.0,7.0,8.0,9.0,
10.0,11.0,12.0,13.0,14.0,15.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,6.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,13.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,4.0,3.0,4.0,
5.0,6.0,7.0,8.0,9.0,10.0,11.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,2.0,1.0,2.0,3.0,
4.0,5.0,6.0,7.0,8.0,9.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,11.0,10.0,9.0,10.0,11.0,
12.0,13.0,14.0,15.0,16.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,9.0,8.0,7.0,8.0,
9.0,10.0,11.0,12.0,13.0,14.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,7.0,6.0,5.0,6.0,
7.0,8.0,9.0,10.0,11.0,12.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,5.0,4.0,3.0,4.0,5.0,
6.0,7.0,8.0,9.0,10.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,
6.0,7.0,8.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,
14.0,15.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,
11.0,12.0,13.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,

```
10.0,11.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,
5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,3.0,2.0,
1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,13.0,12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,14.0,12.0,11.0,
10.0,9.0,8.0,9.0,10.0,11.0,12.0,13.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,12.0,10.0,9.0,
8.0,7.0,6.0,7.0,8.0,9.0,10.0,11.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,10.0,8.0,7.0,6.0,5.0,
4.0,5.0,6.0,7.0,8.0,9.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,8.0,6.0,5.0,4.0,3.0,2.0,3.0,
4.0,5.0,6.0,7.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,6.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,
4.0,5.0,14.0,13.0,12.0,11.0,10.0,9.0,10.0,11.0,12.0,13.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,
10.0,11.0,12.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,9.0,10.0,11.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,
8.0,9.0,10.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,7.0,8.0,9.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,
8.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,6.0,7.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,6.0,6.0,5.0,
4.0,3.0,2.0,1.0,2.0,3.0,4.0,5.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,4.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,10.0,11.0,12.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,11.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,8.0,9.0,10.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,9.0,11.0,10.0,9.0,8.0,
7.0,6.0,5.0,6.0,7.0,8.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,7.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,
4.0,5.0,6.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,5.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,4.0,
6.0,5.0,4.0,3.0,2.0,1.0,0.0,1.0,2.0,3.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,10.0,11.0,
15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,9.0,10.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,
9.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,8.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,
7.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,6.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,5.0,9.0,
8.0,7.0,6.0,5.0,4.0,3.0,2.0,3.0,4.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,3.0,7.0,6.0,5.0,4.0,
3.0,2.0,1.0,0.0,1.0,2.0,17.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,10.0,16.0,15.0,14.0,
13.0,12.0,11.0,10.0,9.0,8.0,9.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,8.0,14.0,13.0,
12.0,11.0,10.0,9.0,8.0,7.0,6.0,7.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,6.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,5.0,4.0,5.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,4.0,10.0,9.0,8.0,7.0,
6.0,5.0,4.0,3.0,2.0,3.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
1.0,0.0,1.0,18.0,17.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,17.0,16.0,15.0,14.0,13.0,12.0,
11.0,10.0,9.0,8.0,16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,15.0,14.0,13.0,12.0,11.0,
10.0,9.0,8.0,7.0,6.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,13.0,12.0,11.0,10.0,9.0,
8.0,7.0,6.0,5.0,4.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,11.0,10.0,9.0,8.0,7.0,6.0,
5.0,4.0,3.0,2.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,
1.0,0.0),
                        nrow=100,ncol=100, byrow=TRUE)



#Fixing the row and column names to nice labels from first cost matrix
rownames(costmatrix2) <- rownames(costmatrix)
colnames(costmatrix2) <- colnames(costmatrix)
costmatrix2
```

```
##     A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y
## A   0  1  2  3  4  5  6  7  8  9  1  2  3  4  5  6  7  8  9 10  2  3  4  5  6
## B   1  0  1  2  3  4  5  6  7  8  2  1  2  3  4  5  6  7  8  9  3  2  3  4  5
## C   2  1  0  1  2  3  4  5  6  7  3  2  1  2  3  4  5  6  7  8  4  3  2  3  4
## D   3  2  1  0  1  2  3  4  5  6  4  3  2  1  2  3  4  5  6  7  5  4  3  2  3
## E   4  3  2  1  0  1  2  3  4  5  5  4  3  2  1  2  3  4  5  6  6  5  4  3  2
## F   5  4  3  2  1  0  1  2  3  4  6  5  4  3  2  1  2  3  4  5  7  6  5  4  3
## G   6  5  4  3  2  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4  8  7  6  5  4
## H   7  6  5  4  3  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3  9  8  7  6  5
## I   8  7  6  5  4  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2 10  9  8  7  6
## J   9  8  7  6  5  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1 11 10  9  8  7
## K   1  2  3  4  5  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9  1  2  3  4  5
## L   2  1  2  3  4  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8  2  1  2  3  4
```

```
## M   3  2  1  2  3  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7  3  2  1  2  3
## N   4  3  2  1  2  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6  4  3  2  1  2
## O   5  4  3  2  1  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5  5  4  3  2  1
## P   6  5  4  3  2  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4  6  5  4  3  2
## Q   7  6  5  4  3  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3  7  6  5  4  3
## R   8  7  6  5  4  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2  8  7  6  5  4
## S   9  8  7  6  5  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1  9  8  7  6  5
## T  10  9  8  7  6  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0 10  9  8  7  6
## U   2  3  4  5  6  7  8  9 10 11  1  2  3  4  5  6  7  8  9 10  0  1  2  3  4
## V   3  2  3  4  5  6  7  8  9 10  2  1  2  3  4  5  6  7  8  9  1  0  1  2  3
## W   4  3  2  3  4  5  6  7  8  9  3  2  1  2  3  4  5  6  7  8  2  1  0  1  2
## X   5  4  3  2  3  4  5  6  7  8  4  3  2  1  2  3  4  5  6  7  3  2  1  0  1
## Y   6  5  4  3  2  3  4  5  6  7  5  4  3  2  1  2  3  4  5  6  4  3  2  1  0
## Z   7  6  5  4  3  2  3  4  5  6  6  5  4  3  2  1  2  3  4  5  5  4  3  2  1
## AA  8  7  6  5  4  3  2  3  4  5  7  6  5  4  3  2  1  2  3  4  6  5  4  3  2
## AB  9  8  7  6  5  4  3  2  3  4  8  7  6  5  4  3  2  1  2  3  7  6  5  4  3
## AC 10  9  8  7  6  5  4  3  2  3  9  8  7  6  5  4  3  2  1  2  8  7  6  5  4
## AD 11 10  9  8  7  6  5  4  3  2 10  9  8  7  6  5  4  3  2  1  9  8  7  6  5
## AE  3  4  5  6  7  8  9 10 11 12  2  3  4  5  6  7  8  9 10 11  1  2  3  4  5
## AF  4  3  4  5  6  7  8  9 10 11  3  2  3  4  5  6  7  8  9 10  2  1  2  3  4
## AG  5  4  3  4  5  6  7  8  9 10  4  3  2  3  4  5  6  7  8  9  3  2  1  2  3
## AH  6  5  4  3  4  5  6  7  8  9  5  4  3  2  3  4  5  6  7  8  4  3  2  1  2
## AI  7  6  5  4  3  4  5  6  7  8  6  5  4  3  2  3  4  5  6  7  5  4  3  2  1
## AJ  8  7  6  5  4  3  4  5  6  7  7  6  5  4  3  2  3  4  5  6  6  5  4  3  2
## AK  9  8  7  6  5  4  3  4  5  6  8  7  6  5  4  3  2  3  4  5  7  6  5  4  3
## AL 10  9  8  7  6  5  4  3  4  5  9  8  7  6  5  4  3  2  3  4  8  7  6  5  4
## AM 11 10  9  8  7  6  5  4  3  4 10  9  8  7  6  5  4  3  2  3  9  8  7  6  5
## AN 12 11 10  9  8  7  6  5  4  3 11 10  9  8  7  6  5  4  3  2 10  9  8  7  6
## AO  4  5  6  7  8  9 10 11 12 13  3  4  5  6  7  8  9 10 11 12  2  3  4  5  6
## AP  5  4  5  6  7  8  9 10 11 12  4  3  4  5  6  7  8  9 10 11  3  2  3  4  5
## AQ  6  5  4  5  6  7  8  9 10 11  5  4  3  4  5  6  7  8  9 10  4  3  2  3  4
## AR  7  6  5  4  5  6  7  8  9 10  6  5  4  3  4  5  6  7  8  9  5  4  3  2  3
## AS  8  7  6  5  4  5  6  7  8  9  7  6  5  4  3  4  5  6  7  8  6  5  4  3  2
## AT  9  8  7  6  5  4  5  6  7  8  8  7  6  5  4  3  4  5  6  7  7  6  5  4  3
## AU 10  9  8  7  6  5  4  5  6  7  9  8  7  6  5  4  3  4  5  6  8  7  6  5  4
## AV 11 10  9  8  7  6  5  4  5  6 10  9  8  7  6  5  4  3  4  5  9  8  7  6  5
## AW 12 11 10  9  8  7  6  5  4  5 11 10  9  8  7  6  5  4  3  4 10  9  8  7  6
## AX 13 12 11 10  9  8  7  6  5  4 12 11 10  9  8  7  6  5  4  3 11 10  9  8  7
## AY  5  6  7  8  9 10 11 12 13 14  4  5  6  7  8  9 10 11 12 13  3  4  5  6  7
## AZ  6  5  6  7  8  9 10 11 12 13  5  4  5  6  7  8  9 10 11 12  4  3  4  5  6
## BA  7  6  5  6  7  8  9 10 11 12  6  5  4  5  6  7  8  9 10 11  5  4  3  4  5
## BB  8  7  6  5  6  7  8  9 10 11  7  6  5  4  5  6  7  8  9 10  6  5  4  3  4
## BC  9  8  7  6  5  6  7  8  9 10  8  7  6  5  4  5  6  7  8  9  7  6  5  4  3
## BD 10  9  8  7  6  5  6  7  8  9  9  8  7  6  5  4  5  6  7  8  8  7  6  5  4
## BE 11 10  9  8  7  6  5  6  7  8 10  9  8  7  6  5  4  5  6  7  9  8  7  6  5
## BF 12 11 10  9  8  7  6  5  6  7 11 10  9  8  7  6  5  4  5  6 10  9  8  7  6
## BG 13 12 11 10  9  8  7  6  5  6 12 11 10  9  8  7  6  5  4  5 11 10  9  8  7
## BH 14 13 12 11 10  9  8  7  6  5 13 12 11 10  9  8  7  6  5  4 12 11 10  9  8
## BI  6  7  8  9 10 11 12 13 14 15  5  6  7  8  9 10 11 12 13 14  4  5  6  7  8
## BJ  7  6  7  8  9 10 11 12 13 14  6  5  6  7  8  9 10 11 12 13  5  4  5  6  7
## BK  8  7  6  7  8  9 10 11 12 13  7  6  5  6  7  8  9 10 11 12  6  5  4  5  6
## BL  9  8  7  6  7  8  9 10 11 12  8  7  6  5  6  7  8  9 10 11  7  6  5  4  5
## BM 10  9  8  7  6  7  8  9 10 11  9  8  7  6  5  6  7  8  9 10  8  7  6  5  4
## BN 11 10  9  8  7  6  7  8  9 10 10  9  8  7  6  5  6  7  8  9  9  8  7  6  5
```

```
## BO 12 11 10  9  8  7  6  7  8  9 11 10  9  8  7  6  5  6  7  8 10  9  8  7  6
## BP 13 12 11 10  9  8  7  6  7  8 12 11 10  9  8  7  6  5  6  7 11 10  9  8  7
## BQ 14 13 12 11 10  9  8  7  6  7 13 12 11 10  9  8  7  6  5  6 12 11 10  9  8
## BR 15 14 13 12 11 10  9  8  7  6 14 13 12 11 10  9  8  7  6  5 13 12 11 10  9
## BS  7  8  9 10 11 12 13 14 15 16  6  7  8  9 10 11 12 13 14 15  5  6  7  8  9
## BT  8  7  8  9 10 11 12 13 14 15  7  6  7  8  9 10 11 12 13 14  6  5  6  7  8
## BU  9  8  7  8  9 10 11 12 13 14  8  7  6  7  8  9 10 11 12 13  7  6  5  6  7
## BV 10  9  8  7  8  9 10 11 12 13  9  8  7  6  7  8  9 10 11 12  8  7  6  5  6
## BW 11 10  9  8  7  8  9 10 11 12 10  9  8  7  6  7  8  9 10 11  9  8  7  6  5
## BX 12 11 10  9  8  7  8  9 10 11 11 10  9  8  7  6  7  8  9 10 10  9  8  7  6
## BY 13 12 11 10  9  8  7  8  9 10 12 11 10  9  8  7  6  7  8  9 11 10  9  8  7
## BZ 14 13 12 11 10  9  8  7  8  9 13 12 11 10  9  8  7  6  7  8 12 11 10  9  8
## CA 15 14 13 12 11 10  9  8  7  8 14 13 12 11 10  9  8  7  6  7 13 12 11 10  9
## CB 16 15 14 13 12 11 10  9  8  7 15 14 13 12 11 10  9  8  7  6 14 13 12 11 10
## CC  8  9 10 11 12 13 14 15 16 17  7  8  9 10 11 12 13 14 15 16  6  7  8  9 10
## CD  9  8  9 10 11 12 13 14 15 16  8  7  8  9 10 11 12 13 14 15  7  6  7  8  9
## CE 10  9  8  9 10 11 12 13 14 15  9  8  7  8  9 10 11 12 13 14  8  7  6  7  8
## CF 11 10  9  8  9 10 11 12 13 14 10  9  8  7  8  9 10 11 12 13  9  8  7  6  7
## CG 12 11 10  9  8  9 10 11 12 13 11 10  9  8  7  8  9 10 11 12 10  9  8  7  6
## CH 13 12 11 10  9  8  9 10 11 12 12 11 10  9  8  7  8  9 10 11 11 10  9  8  7
## CI 14 13 12 11 10  9  8  9 10 11 13 12 11 10  9  8  7  8  9 10 12 11 10  9  8
## CJ 15 14 13 12 11 10  9  8  9 10 14 13 12 11 10  9  8  7  8  9 13 12 11 10  9
## CK 16 15 14 13 12 11 10  9  8  9 15 14 13 12 11 10  9  8  7  8 14 13 12 11 10
## CL 17 16 15 14 13 12 11 10  9  8 16 15 14 13 12 11 10  9  8  7 15 14 13 12 11
## CM  9 10 11 12 13 14 15 16 17 18  8  9 10 11 12 13 14 15 16 17  7  8  9 10 11
## CN 10  9 10 11 12 13 14 15 16 17  9  8  9 10 11 12 13 14 15 16  8  7  8  9 10
## CO 11 10  9 10 11 12 13 14 15 16 10  9  8  9 10 11 12 13 14 15  9  8  7  8  9
## CP 12 11 10  9 10 11 12 13 14 15 11 10  9  8  9 10 11 12 13 14 10  9  8  7  8
## CQ 13 12 11 10  9 10 11 12 13 14 12 11 10  9  8  9 10 11 12 13 11 10  9  8  7
## CR 14 13 12 11 10  9 10 11 12 13 13 12 11 10  9  8  9 10 11 12 12 11 10  9  8
## CS 15 14 13 12 11 10  9 10 11 12 14 13 12 11 10  9  8  9 10 11 13 12 11 10  9
## CT 16 15 14 13 12 11 10  9 10 11 15 14 13 12 11 10  9  8  9 10 14 13 12 11 10
## CU 17 16 15 14 13 12 11 10  9 10 16 15 14 13 12 11 10  9  8  9 15 14 13 12 11
## CV 18 17 16 15 14 13 12 11 10  9 17 16 15 14 13 12 11 10  9  8 16 15 14 13 12
##     Z AA AB AC AD AE AF AG AH AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX
## A   7  8  9 10 11  3  4  5  6  7  8  9 10 11 12  4  5  6  7  8  9 10 11 12 13
## B   6  7  8  9 10  4  3  4  5  6  7  8  9 10 11  5  4  5  6  7  8  9 10 11 12
## C   5  6  7  8  9  5  4  3  4  5  6  7  8  9 10  6  5  4  5  6  7  8  9 10 11
## D   4  5  6  7  8  6  5  4  3  4  5  6  7  8  9  7  6  5  4  5  6  7  8  9 10
## E   3  4  5  6  7  7  6  5  4  3  4  5  6  7  8  8  7  6  5  4  5  6  7  8  9
## F   2  3  4  5  6  8  7  6  5  4  3  4  5  6  7  9  8  7  6  5  4  5  6  7  8
## G   3  2  3  4  5  9  8  7  6  5  4  3  4  5  6 10  9  8  7  6  5  4  5  6  7
## H   4  3  2  3  4 10  9  8  7  6  5  4  3  4  5 11 10  9  8  7  6  5  4  5  6
## I   5  4  3  2  3 11 10  9  8  7  6  5  4  3  4 12 11 10  9  8  7  6  5  4  5
## J   6  5  4  3  2 12 11 10  9  8  7  6  5  4  3 13 12 11 10  9  8  7  6  5  4
## K   6  7  8  9 10  2  3  4  5  6  7  8  9 10 11  3  4  5  6  7  8  9 10 11 12
## L   5  6  7  8  9  3  2  3  4  5  6  7  8  9 10  4  3  4  5  6  7  8  9 10 11
## M   4  5  6  7  8  4  3  2  3  4  5  6  7  8  9  5  4  3  4  5  6  7  8  9 10
## N   3  4  5  6  7  5  4  3  2  3  4  5  6  7  8  6  5  4  3  4  5  6  7  8  9
## O   2  3  4  5  6  6  5  4  3  2  3  4  5  6  7  7  6  5  4  3  4  5  6  7  8
## P   1  2  3  4  5  7  6  5  4  3  2  3  4  5  6  8  7  6  5  4  3  4  5  6  7
## Q   2  1  2  3  4  8  7  6  5  4  3  2  3  4  5  9  8  7  6  5  4  3  4  5  6
## R   3  2  1  2  3  9  8  7  6  5  4  3  2  3  4 10  9  8  7  6  5  4  3  4  5
## S   4  3  2  1  2 10  9  8  7  6  5  4  3  2  3 11 10  9  8  7  6  5  4  3  4
```

```
## T   5  4  3  2  1 11 10  9  8  7  6  5  4  3  2 12 11 10  9  8  7  6  5  4  3
## U   5  6  7  8  9  1  2  3  4  5  6  7  8  9 10  2  3  4  5  6  7  8  9 10 11
## V   4  5  6  7  8  2  1  2  3  4  5  6  7  8  9  3  2  3  4  5  6  7  8  9 10
## W   3  4  5  6  7  3  2  1  2  3  4  5  6  7  8  4  3  2  3  4  5  6  7  8  9
## X   2  3  4  5  6  4  3  2  1  2  3  4  5  6  7  5  4  3  2  3  4  5  6  7  8
## Y   1  2  3  4  5  5  4  3  2  1  2  3  4  5  6  6  5  4  3  2  3  4  5  6  7
## Z   0  1  2  3  4  6  5  4  3  2  1  2  3  4  5  7  6  5  4  3  2  3  4  5  6
## AA  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4  8  7  6  5  4  3  2  3  4  5
## AB  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3  9  8  7  6  5  4  3  2  3  4
## AC  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2 10  9  8  7  6  5  4  3  2  3
## AD  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1 11 10  9  8  7  6  5  4  3  2
## AE  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9  1  2  3  4  5  6  7  8  9 10
## AF  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8  2  1  2  3  4  5  6  7  8  9
## AG  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7  3  2  1  2  3  4  5  6  7  8
## AH  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6  4  3  2  1  2  3  4  5  6  7
## AI  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5  5  4  3  2  1  2  3  4  5  6
## AJ  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4  6  5  4  3  2  1  2  3  4  5
## AK  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4
## AL  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3
## AM  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2
## AN  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1
## AO  7  8  9 10 11  1  2  3  4  5  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9
## AP  6  7  8  9 10  2  1  2  3  4  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8
## AQ  5  6  7  8  9  3  2  1  2  3  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7
## AR  4  5  6  7  8  4  3  2  1  2  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6
## AS  3  4  5  6  7  5  4  3  2  1  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5
## AT  2  3  4  5  6  6  5  4  3  2  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4
## AU  3  2  3  4  5  7  6  5  4  3  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3
## AV  4  3  2  3  4  8  7  6  5  4  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2
## AW  5  4  3  2  3  9  8  7  6  5  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1
## AX  6  5  4  3  2 10  9  8  7  6  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0
## AY  8  9 10 11 12  2  3  4  5  6  7  8  9 10 11  1  2  3  4  5  6  7  8  9 10
## AZ  7  8  9 10 11  3  2  3  4  5  6  7  8  9 10  2  1  2  3  4  5  6  7  8  9
## BA  6  7  8  9 10  4  3  2  3  4  5  6  7  8  9  3  2  1  2  3  4  5  6  7  8
## BB  5  6  7  8  9  5  4  3  2  3  4  5  6  7  8  4  3  2  1  2  3  4  5  6  7
## BC  4  5  6  7  8  6  5  4  3  2  3  4  5  6  7  5  4  3  2  1  2  3  4  5  6
## BD  3  4  5  6  7  7  6  5  4  3  2  3  4  5  6  6  5  4  3  2  1  2  3  4  5
## BE  4  3  4  5  6  8  7  6  5  4  3  2  3  4  5  7  6  5  4  3  2  1  2  3  4
## BF  5  4  3  4  5  9  8  7  6  5  4  3  2  3  4  8  7  6  5  4  3  2  1  2  3
## BG  6  5  4  3  4 10  9  8  7  6  5  4  3  2  3  9  8  7  6  5  4  3  2  1  2
## BH  7  6  5  4  3 11 10  9  8  7  6  5  4  3  2 10  9  8  7  6  5  4  3  2  1
## BI  9 10 11 12 13  3  4  5  6  7  8  9 10 11 12  2  3  4  5  6  7  8  9 10 11
## BJ  8  9 10 11 12  4  3  4  5  6  7  8  9 10 11  3  2  3  4  5  6  7  8  9 10
## BK  7  8  9 10 11  5  4  3  4  5  6  7  8  9 10  4  3  2  3  4  5  6  7  8  9
## BL  6  7  8  9 10  6  5  4  3  4  5  6  7  8  9  5  4  3  2  3  4  5  6  7  8
## BM  5  6  7  8  9  7  6  5  4  3  4  5  6  7  8  6  5  4  3  2  3  4  5  6  7
## BN  4  5  6  7  8  8  7  6  5  4  3  4  5  6  7  7  6  5  4  3  2  3  4  5  6
## BO  5  4  5  6  7  9  8  7  6  5  4  3  4  5  6  8  7  6  5  4  3  2  3  4  5
## BP  6  5  4  5  6 10  9  8  7  6  5  4  3  4  5  9  8  7  6  5  4  3  2  3  4
## BQ  7  6  5  4  5 11 10  9  8  7  6  5  4  3  4 10  9  8  7  6  5  4  3  2  3
## BR  8  7  6  5  4 12 11 10  9  8  7  6  5  4  3 11 10  9  8  7  6  5  4  3  2
## BS 10 11 12 13 14  4  5  6  7  8  9 10 11 12 13  3  4  5  6  7  8  9 10 11 12
## BT  9 10 11 12 13  5  4  5  6  7  8  9 10 11 12  4  3  4  5  6  7  8  9 10 11
## BU  8  9 10 11 12  6  5  4  5  6  7  8  9 10 11  5  4  3  4  5  6  7  8  9 10
```

```
## BV  7  8  9 10 11  7  6  5  4  5  6  7  8  9 10  6  5  4  3  4  5  6  7  8  9
## BW  6  7  8  9 10  8  7  6  5  4  5  6  7  8  9  7  6  5  4  3  4  5  6  7  8
## BX  5  6  7  8  9  9  8  7  6  5  4  5  6  7  8  8  7  6  5  4  3  4  5  6  7
## BY  6  5  6  7  8 10  9  8  7  6  5  4  5  6  7  9  8  7  6  5  4  3  4  5  6
## BZ  7  6  5  6  7 11 10  9  8  7  6  5  4  5  6 10  9  8  7  6  5  4  3  4  5
## CA  8  7  6  5  6 12 11 10  9  8  7  6  5  4  5 11 10  9  8  7  6  5  4  3  4
## CB  9  8  7  6  5 13 12 11 10  9  8  7  6  5  4 12 11 10  9  8  7  6  5  4  3
## CC 11 12 13 14 15  5  6  7  8  9 10 11 12 13 14  4  5  6  7  8  9 10 11 12 13
## CD 10 11 12 13 14  6  5  6  7  8  9 10 11 12 13  5  4  5  6  7  8  9 10 11 12
## CE  9 10 11 12 13  7  6  5  6  7  8  9 10 11 12  6  5  4  5  6  7  8  9 10 11
## CF  8  9 10 11 12  8  7  6  5  6  7  8  9 10 11  7  6  5  4  5  6  7  8  9 10
## CG  7  8  9 10 11  9  8  7  6  5  6  7  8  9 10  8  7  6  5  4  5  6  7  8  9
## CH  6  7  8  9 10 10  9  8  7  6  5  6  7  8  9  9  8  7  6  5  4  5  6  7  8
## CI  7  6  7  8  9 11 10  9  8  7  6  5  6  7  8 10  9  8  7  6  5  4  5  6  7
## CJ  8  7  6  7  8 12 11 10  9  8  7  6  5  6  7 11 10  9  8  7  6  5  4  5  6
## CK  9  8  7  6  7 13 12 11 10  9  8  7  6  5  6 12 11 10  9  8  7  6  5  4  5
## CL 10  9  8  7  6 14 13 12 11 10  9  8  7  6  5 13 12 11 10  9  8  7  6  5  4
## CM 12 13 14 15 16  6  7  8  9 10 11 12 13 14 15  5  6  7  8  9 10 11 12 13 14
## CN 11 12 13 14 15  7  6  7  8  9 10 11 12 13 14  6  5  6  7  8  9 10 11 12 13
## CO 10 11 12 13 14  8  7  6  7  8  9 10 11 12 13  7  6  5  6  7  8  9 10 11 12
## CP  9 10 11 12 13  9  8  7  6  7  8  9 10 11 12  8  7  6  5  6  7  8  9 10 11
## CQ  8  9 10 11 12 10  9  8  7  6  7  8  9 10 11  9  8  7  6  5  6  7  8  9 10
## CR  7  8  9 10 11 11 10  9  8  7  6  7  8  9 10 10  9  8  7  6  5  6  7  8  9
## CS  8  7  8  9 10 12 11 10  9  8  7  6  7  8  9 11 10  9  8  7  6  5  6  7  8
## CT  9  8  7  8  9 13 12 11 10  9  8  7  6  7  8 12 11 10  9  8  7  6  5  6  7
## CU 10  9  8  7  8 14 13 12 11 10  9  8  7  6  7 13 12 11 10  9  8  7  6  5  6
## CV 11 10  9  8  7 15 14 13 12 11 10  9  8  7  6 14 13 12 11 10  9  8  7  6  5
##    AY AZ BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW
## A   5  6  7  8  9 10 11 12 13 14  6  7  8  9 10 11 12 13 14 15  7  8  9 10 11
## B   6  5  6  7  8  9 10 11 12 13  7  6  7  8  9 10 11 12 13 14  8  7  8  9 10
## C   7  6  5  6  7  8  9 10 11 12  8  7  6  7  8  9 10 11 12 13  9  8  7  8  9
## D   8  7  6  5  6  7  8  9 10 11  9  8  7  6  7  8  9 10 11 12 10  9  8  7  8
## E   9  8  7  6  5  6  7  8  9 10 10  9  8  7  6  7  8  9 10 11 11 10  9  8  7
## F  10  9  8  7  6  5  6  7  8  9 11 10  9  8  7  6  7  8  9 10 12 11 10  9  8
## G  11 10  9  8  7  6  5  6  7  8 12 11 10  9  8  7  6  7  8  9 13 12 11 10  9
## H  12 11 10  9  8  7  6  5  6  7 13 12 11 10  9  8  7  6  7  8 14 13 12 11 10
## I  13 12 11 10  9  8  7  6  5  6 14 13 12 11 10  9  8  7  6  7 15 14 13 12 11
## J  14 13 12 11 10  9  8  7  6  5 15 14 13 12 11 10  9  8  7  6 16 15 14 13 12
## K   4  5  6  7  8  9 10 11 12 13  5  6  7  8  9 10 11 12 13 14  6  7  8  9 10
## L   5  4  5  6  7  8  9 10 11 12  6  5  6  7  8  9 10 11 12 13  7  6  7  8  9
## M   6  5  4  5  6  7  8  9 10 11  7  6  5  6  7  8  9 10 11 12  8  7  6  7  8
## N   7  6  5  4  5  6  7  8  9 10  8  7  6  5  6  7  8  9 10 11  9  8  7  6  7
## O   8  7  6  5  4  5  6  7  8  9  9  8  7  6  5  6  7  8  9 10 10  9  8  7  6
## P   9  8  7  6  5  4  5  6  7  8 10  9  8  7  6  5  6  7  8  9 11 10  9  8  7
## Q  10  9  8  7  6  5  4  5  6  7 11 10  9  8  7  6  5  6  7  8 12 11 10  9  8
## R  11 10  9  8  7  6  5  4  5  6 12 11 10  9  8  7  6  5  6  7 13 12 11 10  9
## S  12 11 10  9  8  7  6  5  4  5 13 12 11 10  9  8  7  6  5  6 14 13 12 11 10
## T  13 12 11 10  9  8  7  6  5  4 14 13 12 11 10  9  8  7  6  5 15 14 13 12 11
## U   3  4  5  6  7  8  9 10 11 12  4  5  6  7  8  9 10 11 12 13  5  6  7  8  9
## V   4  3  4  5  6  7  8  9 10 11  5  4  5  6  7  8  9 10 11 12  6  5  6  7  8
## W   5  4  3  4  5  6  7  8  9 10  6  5  4  5  6  7  8  9 10 11  7  6  5  6  7
## X   6  5  4  3  4  5  6  7  8  9  7  6  5  4  5  6  7  8  9 10  8  7  6  5  6
## Y   7  6  5  4  3  4  5  6  7  8  8  7  6  5  4  5  6  7  8  9  9  8  7  6  5
## Z   8  7  6  5  4  3  4  5  6  7  9  8  7  6  5  4  5  6  7  8 10  9  8  7  6
```

```
## AA  9  8  7  6  5  4  3  4  5  6 10  9  8  7  6  5  4  5  6  7 11 10  9  8  7
## AB 10  9  8  7  6  5  4  3  4  5 11 10  9  8  7  6  5  4  5  6 12 11 10  9  8
## AC 11 10  9  8  7  6  5  4  3  4 12 11 10  9  8  7  6  5  4  5 13 12 11 10  9
## AD 12 11 10  9  8  7  6  5  4  3 13 12 11 10  9  8  7  6  5  4 14 13 12 11 10
## AE  2  3  4  5  6  7  8  9 10 11  3  4  5  6  7  8  9 10 11 12  4  5  6  7  8
## AF  3  2  3  4  5  6  7  8  9 10  4  3  4  5  6  7  8  9 10 11  5  4  5  6  7
## AG  4  3  2  3  4  5  6  7  8  9  5  4  3  4  5  6  7  8  9 10  6  5  4  5  6
## AH  5  4  3  2  3  4  5  6  7  8  6  5  4  3  4  5  6  7  8  9  7  6  5  4  5
## AI  6  5  4  3  2  3  4  5  6  7  7  6  5  4  3  4  5  6  7  8  8  7  6  5  4
## AJ  7  6  5  4  3  2  3  4  5  6  8  7  6  5  4  3  4  5  6  7  9  8  7  6  5
## AK  8  7  6  5  4  3  2  3  4  5  9  8  7  6  5  4  3  4  5  6 10  9  8  7  6
## AL  9  8  7  6  5  4  3  2  3  4 10  9  8  7  6  5  4  3  4  5 11 10  9  8  7
## AM 10  9  8  7  6  5  4  3  2  3 11 10  9  8  7  6  5  4  3  4 12 11 10  9  8
## AN 11 10  9  8  7  6  5  4  3  2 12 11 10  9  8  7  6  5  4  3 13 12 11 10  9
## AO  1  2  3  4  5  6  7  8  9 10  2  3  4  5  6  7  8  9 10 11  3  4  5  6  7
## AP  2  1  2  3  4  5  6  7  8  9  3  2  3  4  5  6  7  8  9 10  4  3  4  5  6
## AQ  3  2  1  2  3  4  5  6  7  8  4  3  2  3  4  5  6  7  8  9  5  4  3  4  5
## AR  4  3  2  1  2  3  4  5  6  7  5  4  3  2  3  4  5  6  7  8  6  5  4  3  4
## AS  5  4  3  2  1  2  3  4  5  6  6  5  4  3  2  3  4  5  6  7  7  6  5  4  3
## AT  6  5  4  3  2  1  2  3  4  5  7  6  5  4  3  2  3  4  5  6  8  7  6  5  4
## AU  7  6  5  4  3  2  1  2  3  4  8  7  6  5  4  3  2  3  4  5  9  8  7  6  5
## AV  8  7  6  5  4  3  2  1  2  3  9  8  7  6  5  4  3  2  3  4 10  9  8  7  6
## AW  9  8  7  6  5  4  3  2  1  2 10  9  8  7  6  5  4  3  2  3 11 10  9  8  7
## AX 10  9  8  7  6  5  4  3  2  1 11 10  9  8  7  6  5  4  3  2 12 11 10  9  8
## AY  0  1  2  3  4  5  6  7  8  9  1  2  3  4  5  6  7  8  9 10  2  3  4  5  6
## AZ  1  0  1  2  3  4  5  6  7  8  2  1  2  3  4  5  6  7  8  9  3  2  3  4  5
## BA  2  1  0  1  2  3  4  5  6  7  3  2  1  2  3  4  5  6  7  8  4  3  2  3  4
## BB  3  2  1  0  1  2  3  4  5  6  4  3  2  1  2  3  4  5  6  7  5  4  3  2  3
## BC  4  3  2  1  0  1  2  3  4  5  5  4  3  2  1  2  3  4  5  6  6  5  4  3  2
## BD  5  4  3  2  1  0  1  2  3  4  6  5  4  3  2  1  2  3  4  5  7  6  5  4  3
## BE  6  5  4  3  2  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4  8  7  6  5  4
## BF  7  6  5  4  3  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3  9  8  7  6  5
## BG  8  7  6  5  4  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2 10  9  8  7  6
## BH  9  8  7  6  5  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1 11 10  9  8  7
## BI  1  2  3  4  5  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9  1  2  3  4  5
## BJ  2  1  2  3  4  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8  2  1  2  3  4
## BK  3  2  1  2  3  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7  3  2  1  2  3
## BL  4  3  2  1  2  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6  4  3  2  1  2
## BM  5  4  3  2  1  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5  5  4  3  2  1
## BN  6  5  4  3  2  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4  6  5  4  3  2
## BO  7  6  5  4  3  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3  7  6  5  4  3
## BP  8  7  6  5  4  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2  8  7  6  5  4
## BQ  9  8  7  6  5  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1  9  8  7  6  5
## BR 10  9  8  7  6  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0 10  9  8  7  6
## BS  2  3  4  5  6  7  8  9 10 11  1  2  3  4  5  6  7  8  9 10  0  1  2  3  4
## BT  3  2  3  4  5  6  7  8  9 10  2  1  2  3  4  5  6  7  8  9  1  0  1  2  3
## BU  4  3  2  3  4  5  6  7  8  9  3  2  1  2  3  4  5  6  7  8  2  1  0  1  2
## BV  5  4  3  2  3  4  5  6  7  8  4  3  2  1  2  3  4  5  6  7  3  2  1  0  1
## BW  6  5  4  3  2  3  4  5  6  7  5  4  3  2  1  2  3  4  5  6  4  3  2  1  0
## BX  7  6  5  4  3  2  3  4  5  6  6  5  4  3  2  1  2  3  4  5  5  4  3  2  1
## BY  8  7  6  5  4  3  2  3  4  5  7  6  5  4  3  2  1  2  3  4  6  5  4  3  2
## BZ  9  8  7  6  5  4  3  2  3  4  8  7  6  5  4  3  2  1  2  3  7  6  5  4  3
## CA 10  9  8  7  6  5  4  3  2  3  9  8  7  6  5  4  3  2  1  2  8  7  6  5  4
## CB 11 10  9  8  7  6  5  4  3  2 10  9  8  7  6  5  4  3  2  1  9  8  7  6  5
```

```
## CC  3  4  5  6  7  8  9 10 11 12  2  3  4  5  6  7  8  9 10 11  1  2  3  4  5
## CD  4  3  4  5  6  7  8  9 10 11  3  2  3  4  5  6  7  8  9 10  2  1  2  3  4
## CE  5  4  3  4  5  6  7  8  9 10  4  3  2  3  4  5  6  7  8  9  3  2  1  2  3
## CF  6  5  4  3  4  5  6  7  8  9  5  4  3  2  3  4  5  6  7  8  4  3  2  1  2
## CG  7  6  5  4  3  4  5  6  7  8  6  5  4  3  2  3  4  5  6  7  5  4  3  2  1
## CH  8  7  6  5  4  3  4  5  6  7  7  6  5  4  3  2  3  4  5  6  6  5  4  3  2
## CI  9  8  7  6  5  4  3  4  5  6  8  7  6  5  4  3  2  3  4  5  7  6  5  4  3
## CJ 10  9  8  7  6  5  4  3  4  5  9  8  7  6  5  4  3  2  3  4  8  7  6  5  4
## CK 11 10  9  8  7  6  5  4  3  4 10  9  8  7  6  5  4  3  2  3  9  8  7  6  5
## CL 12 11 10  9  8  7  6  5  4  3 11 10  9  8  7  6  5  4  3  2 10  9  8  7  6
## CM  4  5  6  7  8  9 10 11 12 13  3  4  5  6  7  8  9 10 11 12  2  3  4  5  6
## CN  5  4  5  6  7  8  9 10 11 12  4  3  4  5  6  7  8  9 10 11  3  2  3  4  5
## CO  6  5  4  5  6  7  8  9 10 11  5  4  3  4  5  6  7  8  9 10  4  3  2  3  4
## CP  7  6  5  4  5  6  7  8  9 10  6  5  4  3  4  5  6  7  8  9  5  4  3  2  3
## CQ  8  7  6  5  4  5  6  7  8  9  7  6  5  4  3  4  5  6  7  8  6  5  4  3  2
## CR  9  8  7  6  5  4  5  6  7  8  8  7  6  5  4  3  4  5  6  7  7  6  5  4  3
## CS 10  9  8  7  6  5  4  5  6  7  9  8  7  6  5  4  3  4  5  6  8  7  6  5  4
## CT 11 10  9  8  7  6  5  4  5  6 10  9  8  7  6  5  4  3  4  5  9  8  7  6  5
## CU 12 11 10  9  8  7  6  5  4  5 11 10  9  8  7  6  5  4  3  4 10  9  8  7  6
## CV 13 12 11 10  9  8  7  6  5  4 12 11 10  9  8  7  6  5  4  3 11 10  9  8  7
##    BX BY BZ CA CB CC CD CE CF CG CH CI CJ CK CL CM CN CO CP CQ CR CS CT CU CV
## A  12 13 14 15 16  8  9 10 11 12 13 14 15 16 17  9 10 11 12 13 14 15 16 17 18
## B  11 12 13 14 15  9  8  9 10 11 12 13 14 15 16 10  9 10 11 12 13 14 15 16 17
## C  10 11 12 13 14 10  9  8  9 10 11 12 13 14 15 11 10  9 10 11 12 13 14 15 16
## D   9 10 11 12 13 11 10  9  8  9 10 11 12 13 14 12 11 10  9 10 11 12 13 14 15
## E   8  9 10 11 12 12 11 10  9  8  9 10 11 12 13 13 12 11 10  9 10 11 12 13 14
## F   7  8  9 10 11 13 12 11 10  9  8  9 10 11 12 14 13 12 11 10  9 10 11 12 13
## G   8  7  8  9 10 14 13 12 11 10  9  8  9 10 11 15 14 13 12 11 10  9 10 11 12
## H   9  8  7  8  9 15 14 13 12 11 10  9  8  9 10 16 15 14 13 12 11 10  9 10 11
## I  10  9  8  7  8 16 15 14 13 12 11 10  9  8  9 17 16 15 14 13 12 11 10  9 10
## J  11 10  9  8  7 17 16 15 14 13 12 11 10  9  8 18 17 16 15 14 13 12 11 10  9
## K  11 12 13 14 15  7  8  9 10 11 12 13 14 15 16  8  9 10 11 12 13 14 15 16 17
## L  10 11 12 13 14  8  7  8  9 10 11 12 13 14 15  9  8  9 10 11 12 13 14 15 16
## M   9 10 11 12 13  9  8  7  8  9 10 11 12 13 14 10  9  8  9 10 11 12 13 14 15
## N   8  9 10 11 12 10  9  8  7  8  9 10 11 12 13 11 10  9  8  9 10 11 12 13 14
## O   7  8  9 10 11 11 10  9  8  7  8  9 10 11 12 12 11 10  9  8  9 10 11 12 13
## P   6  7  8  9 10 12 11 10  9  8  7  8  9 10 11 13 12 11 10  9  8  9 10 11 12
## Q   7  6  7  8  9 13 12 11 10  9  8  7  8  9 10 14 13 12 11 10  9  8  9 10 11
## R   8  7  6  7  8 14 13 12 11 10  9  8  7  8  9 15 14 13 12 11 10  9  8  9 10
## S   9  8  7  6  7 15 14 13 12 11 10  9  8  7  8 16 15 14 13 12 11 10  9  8  9
## T  10  9  8  7  6 16 15 14 13 12 11 10  9  8  7 17 16 15 14 13 12 11 10  9  8
## U  10 11 12 13 14  6  7  8  9 10 11 12 13 14 15  7  8  9 10 11 12 13 14 15 16
## V   9 10 11 12 13  7  6  7  8  9 10 11 12 13 14  8  7  8  9 10 11 12 13 14 15
## W   8  9 10 11 12  8  7  6  7  8  9 10 11 12 13  9  8  7  8  9 10 11 12 13 14
## X   7  8  9 10 11  9  8  7  6  7  8  9 10 11 12 10  9  8  7  8  9 10 11 12 13
## Y   6  7  8  9 10 10  9  8  7  6  7  8  9 10 11 11 10  9  8  7  8  9 10 11 12
## Z   5  6  7  8  9 11 10  9  8  7  6  7  8  9 10 12 11 10  9  8  7  8  9 10 11
## AA  6  5  6  7  8 12 11 10  9  8  7  6  7  8  9 13 12 11 10  9  8  7  8  9 10
## AB  7  6  5  6  7 13 12 11 10  9  8  7  6  7  8 14 13 12 11 10  9  8  7  8  9
## AC  8  7  6  5  6 14 13 12 11 10  9  8  7  6  7 15 14 13 12 11 10  9  8  7  8
## AD  9  8  7  6  5 15 14 13 12 11 10  9  8  7  6 16 15 14 13 12 11 10  9  8  7
## AE  9 10 11 12 13  5  6  7  8  9 10 11 12 13 14  6  7  8  9 10 11 12 13 14 15
## AF  8  9 10 11 12  6  5  6  7  8  9 10 11 12 13  7  6  7  8  9 10 11 12 13 14
## AG  7  8  9 10 11  7  6  5  6  7  8  9 10 11 12  8  7  6  7  8  9 10 11 12 13
```

```
## AH  6  7  8  9 10  8  7  6  5  6  7  8  9 10 11  9  8  7  6  7  8  9 10 11 12
## AI  5  6  7  8  9  9  8  7  6  5  6  7  8  9 10 10  9  8  7  6  7  8  9 10 11
## AJ  4  5  6  7  8 10  9  8  7  6  5  6  7  8  9 11 10  9  8  7  6  7  8  9 10
## AK  5  4  5  6  7 11 10  9  8  7  6  5  6  7  8 12 11 10  9  8  7  6  7  8  9
## AL  6  5  4  5  6 12 11 10  9  8  7  6  5  6  7 13 12 11 10  9  8  7  6  7  8
## AM  7  6  5  4  5 13 12 11 10  9  8  7  6  5  6 14 13 12 11 10  9  8  7  6  7
## AN  8  7  6  5  4 14 13 12 11 10  9  8  7  6  5 15 14 13 12 11 10  9  8  7  6
## AO  8  9 10 11 12  4  5  6  7  8  9 10 11 12 13  5  6  7  8  9 10 11 12 13 14
## AP  7  8  9 10 11  5  4  5  6  7  8  9 10 11 12  6  5  6  7  8  9 10 11 12 13
## AQ  6  7  8  9 10  6  5  4  5  6  7  8  9 10 11  7  6  5  6  7  8  9 10 11 12
## AR  5  6  7  8  9  7  6  5  4  5  6  7  8  9 10  8  7  6  5  6  7  8  9 10 11
## AS  4  5  6  7  8  8  7  6  5  4  5  6  7  8  9  9  8  7  6  5  6  7  8  9 10
## AT  3  4  5  6  7  9  8  7  6  5  4  5  6  7  8 10  9  8  7  6  5  6  7  8  9
## AU  4  3  4  5  6 10  9  8  7  6  5  4  5  6  7 11 10  9  8  7  6  5  6  7  8
## AV  5  4  3  4  5 11 10  9  8  7  6  5  4  5  6 12 11 10  9  8  7  6  5  6  7
## AW  6  5  4  3  4 12 11 10  9  8  7  6  5  4  5 13 12 11 10  9  8  7  6  5  6
## AX  7  6  5  4  3 13 12 11 10  9  8  7  6  5  4 14 13 12 11 10  9  8  7  6  5
## AY  7  8  9 10 11  3  4  5  6  7  8  9 10 11 12  4  5  6  7  8  9 10 11 12 13
## AZ  6  7  8  9 10  4  3  4  5  6  7  8  9 10 11  5  4  5  6  7  8  9 10 11 12
## BA  5  6  7  8  9  5  4  3  4  5  6  7  8  9 10  6  5  4  5  6  7  8  9 10 11
## BB  4  5  6  7  8  6  5  4  3  4  5  6  7  8  9  7  6  5  4  5  6  7  8  9 10
## BC  3  4  5  6  7  7  6  5  4  3  4  5  6  7  8  8  7  6  5  4  5  6  7  8  9
## BD  2  3  4  5  6  8  7  6  5  4  3  4  5  6  7  9  8  7  6  5  4  5  6  7  8
## BE  3  2  3  4  5  9  8  7  6  5  4  3  4  5  6 10  9  8  7  6  5  4  5  6  7
## BF  4  3  2  3  4 10  9  8  7  6  5  4  3  4  5 11 10  9  8  7  6  5  4  5  6
## BG  5  4  3  2  3 11 10  9  8  7  6  5  4  3  4 12 11 10  9  8  7  6  5  4  5
## BH  6  5  4  3  2 12 11 10  9  8  7  6  5  4  3 13 12 11 10  9  8  7  6  5  4
## BI  6  7  8  9 10  2  3  4  5  6  7  8  9 10 11  3  4  5  6  7  8  9 10 11 12
## BJ  5  6  7  8  9  3  2  3  4  5  6  7  8  9 10  4  3  4  5  6  7  8  9 10 11
## BK  4  5  6  7  8  4  3  2  3  4  5  6  7  8  9  5  4  3  4  5  6  7  8  9 10
## BL  3  4  5  6  7  5  4  3  2  3  4  5  6  7  8  6  5  4  3  4  5  6  7  8  9
## BM  2  3  4  5  6  6  5  4  3  2  3  4  5  6  7  7  6  5  4  3  4  5  6  7  8
## BN  1  2  3  4  5  7  6  5  4  3  2  3  4  5  6  8  7  6  5  4  3  4  5  6  7
## BO  2  1  2  3  4  8  7  6  5  4  3  2  3  4  5  9  8  7  6  5  4  3  4  5  6
## BP  3  2  1  2  3  9  8  7  6  5  4  3  2  3  4 10  9  8  7  6  5  4  3  4  5
## BQ  4  3  2  1  2 10  9  8  7  6  5  4  3  2  3 11 10  9  8  7  6  5  4  3  4
## BR  5  4  3  2  1 11 10  9  8  7  6  5  4  3  2 12 11 10  9  8  7  6  5  4  3
## BS  5  6  7  8  9  1  2  3  4  5  6  7  8  9 10  2  3  4  5  6  7  8  9 10 11
## BT  4  5  6  7  8  2  1  2  3  4  5  6  7  8  9  3  2  3  4  5  6  7  8  9 10
## BU  3  4  5  6  7  3  2  1  2  3  4  5  6  7  8  4  3  2  3  4  5  6  7  8  9
## BV  2  3  4  5  6  4  3  2  1  2  3  4  5  6  7  5  4  3  2  3  4  5  6  7  8
## BW  1  2  3  4  5  5  4  3  2  1  2  3  4  5  6  6  5  4  3  2  3  4  5  6  7
## BX  0  1  2  3  4  6  5  4  3  2  1  2  3  4  5  7  6  5  4  3  2  3  4  5  6
## BY  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4  8  7  6  5  4  3  2  3  4  5
## BZ  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3  9  8  7  6  5  4  3  2  3  4
## CA  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2 10  9  8  7  6  5  4  3  2  3
## CB  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1 11 10  9  8  7  6  5  4  3  2
## CC  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9  1  2  3  4  5  6  7  8  9 10
## CD  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8  2  1  2  3  4  5  6  7  8  9
## CE  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7  3  2  1  2  3  4  5  6  7  8
## CF  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6  4  3  2  1  2  3  4  5  6  7
## CG  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5  5  4  3  2  1  2  3  4  5  6
## CH  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4  6  5  4  3  2  1  2  3  4  5
## CI  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3  7  6  5  4  3  2  1  2  3  4
```

```
## CJ  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2  8  7  6  5  4  3  2  1  2  3
## CK  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1  9  8  7  6  5  4  3  2  1  2
## CL  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0 10  9  8  7  6  5  4  3  2  1
## CM  7  8  9 10 11  1  2  3  4  5  6  7  8  9 10  0  1  2  3  4  5  6  7  8  9
## CN  6  7  8  9 10  2  1  2  3  4  5  6  7  8  9  1  0  1  2  3  4  5  6  7  8
## CO  5  6  7  8  9  3  2  1  2  3  4  5  6  7  8  2  1  0  1  2  3  4  5  6  7
## CP  4  5  6  7  8  4  3  2  1  2  3  4  5  6  7  3  2  1  0  1  2  3  4  5  6
## CQ  3  4  5  6  7  5  4  3  2  1  2  3  4  5  6  4  3  2  1  0  1  2  3  4  5
## CR  2  3  4  5  6  6  5  4  3  2  1  2  3  4  5  5  4  3  2  1  0  1  2  3  4
## CS  3  2  3  4  5  7  6  5  4  3  2  1  2  3  4  6  5  4  3  2  1  0  1  2  3
## CT  4  3  2  3  4  8  7  6  5  4  3  2  1  2  3  7  6  5  4  3  2  1  0  1  2
## CU  5  4  3  2  3  9  8  7  6  5  4  3  2  1  2  8  7  6  5  4  3  2  1  0  1
## CV  6  5  4  3  2 10  9  8  7  6  5  4  3  2  1  9  8  7  6  5  4  3  2  1  0
```

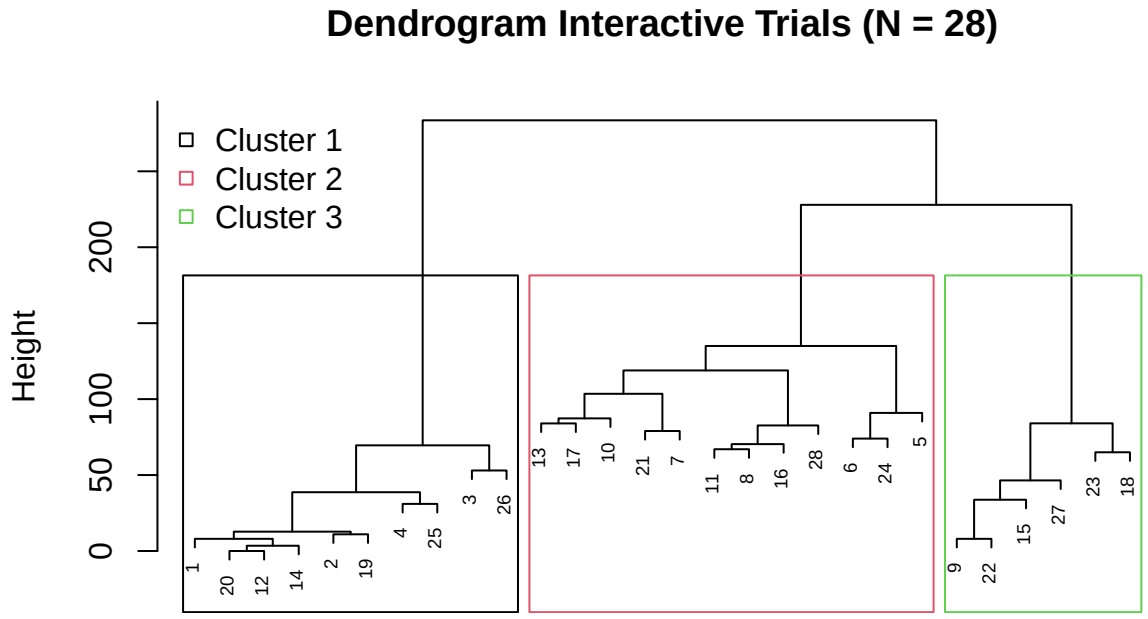
```
#Optimal matching technique for sequence analysis.

#seqdist is a function in TraMineR
#input: sequence object (created above), method (optimal matching), insertion/deletion
#cost (1), cost matrix, missingness
dist.om <- seqdist(aggression.seq, method = "OM", indel = 1.0, sm = costmatrix2,
                   with.missing = TRUE)
```

```
##  [!!] seqdist: 'with.missing' set as FALSE as 'seqdata' has no non-void missing values

##  [>] 28 sequences with 100 distinct states

##  [>] checking 'sm' (size and triangle inequality)

## Warning: At least one substitution cost greater than twice the indel cost. Such
## substitution costs will never be used.

##  [>] 27 distinct  sequences

##  [>] min/max sequence lengths: 59/59

##  [>] computing distances using the OM metric

##  [>] elapsed time: 0.66 secs
```

4. Cluster Determination. We adjusted the dendrogram plot to visualize it with cluster borders and labels for three identified clusters. The legend displays cluster numbers for easy interpretation.

```
#Finding clusters.

#we use hieararchical cluster analysis with Ward's method and the corresponding dendrogram
#to choose an appropriate number of clusters, but other methods to find underlying
#clusters are possible (e.g., latent mixture modeling)
clusterward1 <- agnes(dist.om, diss = TRUE, method = "ward")

# Calculate labels for dendrogram visualization by dividing the order of dyads by the
#number of observations
```

```
label_dend = ((as.integer(clusterward1$order.lab)-1)/max(data_TAP$time))+1

# Convert clusterward1 to hclust object for further manipulation
clust.hclust2 <- as.hclust (clusterward1)

# Perform cutree to extract cluster assignments with k = 3 clusters
groups2 <- cutree (clusterward1, k = 3)

# Reorder the groups based on the dendrogram order
group2.order <- groups2[clust.hclust2$order]

# Extract unique cluster numbers
group2.in.cluster <- unique (group2.order)

# Visualize the dendrogram with cluster borders and labels
plot(clust.hclust2,  labels = label_dend, cex = 0.6, hang = 0.02,
     main = "Dendrogram Interactive Trials (N = 28)",  xlab= "", sub = "")
rect.hclust (clust.hclust2, border = group2.in.cluster, k = 3)
# Add a legend to indicate cluster numbers
legend ('topleft', legend = paste ('Cluster', 1:3), pch = 22, col = 1:3, bty = 'n')
```



We plotted the cluster sequences, each labeled with a name corresponding to the depicted pattern.

```
#Trying cut points.

#3 Types
```

```
cl.3 <- cutree(clusterward1, k = 3) #cutting dendrogram (or tree) by the number of
#determined groups (in this case, 3)
cl.3fac <- factor(cl.3, labels = paste("Type", 1:3)) #turning cut points into a factor
#variable and labeling them

#seqplot is a function in TraMineR
#insert: sequence object, grouping/cut points (in factor form), type of plot (several
#options here, but we could with "I"), sorting criteria, legend, border

seqplot(aggression.seq, group = cl.3fac, type="I",sortv = "from.start",with.legend = F,
        border = NA, main=c("Both low provocation","Mixed provocation","Both high provocation",""))
```