test

2022-11-06

```r
#test4
# df = read.csv("amazon_review_polarity_csv/train.csv", header = FALSE)
```

```r
set.seed(1)
N = 100000
N_t = 0.8*N
reviews_text<-readLines("amazon_review_polarity_csv/train.csv", n = N)
reviews_text<-data.frame(reviews_text)
```

```r
library(tidyr)
reviews_text<-separate(data = reviews_text, col = reviews_text, into = c("Sentiment",
                              "SentimentText"), sep = 4)
```

```r
# Retaining only alphanumeric values in the sentiment column
reviews_text$Sentiment<-gsub("[^[:alnum:] ]","",reviews_text$Sentiment)
# Retaining only alphanumeric values in the sentiment text
reviews_text$SentimentText<-gsub("[^[:alnum:] ]"," ",reviews_text$SentimentText)
# Replacing multiple spaces in the text with single space
reviews_text$SentimentText<-gsub("(?<=[\\s])\\s*|^\\s+|\\s+$", "", reviews_text$SentimentText,
                                 perl=TRUE)
# Writing the output to a file that can be consumed in other projects
write.table(reviews_text,file = "Sentiment Analysis Dataset.csv",row.names = F,col.names = T,sep=',')
```

```r
reviews_text<-readLines('amazon_review_polarity_csv/train.csv', n = N)
# Basic EDA to confirm that the data is read correctly
print(class(reviews_text))
```

```
## [1] "character"
```

```r
print(length(reviews_text))
```

```
## [1] 100000
```

```r
# print(head(reviews_text,2))
# Replacing the positive sentiment value 2 with __label__2
reviews_text<-gsub("\\\"2\\\",","__label__2 ",reviews_text)
# Replacing the negative sentiment value 1 with __label__1
reviews_text<-gsub("\\\"1\\\",","__label__1 ",reviews_text)
# Removing the unnecessary \" characters
reviews_text<-gsub("\\\""," ",reviews_text)
# Replacing multiple spaces in the text with single space
```

```
reviews_text<-gsub("(?<=[\\s])\\s*|^\\s+|\\s+$", "", reviews_text, perl=TRUE)
# Basic EDA post the required processing to confirm input is as desired
print("EDA POST PROCESSING")
```

```
## [1] "EDA POST PROCESSING"
```

```
print(class(reviews_text))
```

```
## [1] "character"
```

```
print(length(reviews_text))
```

```
## [1] 100000
```

```
# print(head(reviews_text,2))
# Writing the revamped file to the directory so we could use it with
# fastText sentiment analyzer project
fileConn<-file("Sentiment Analysis Dataset_ft.txt")
writeLines(reviews_text, fileConn)
close(fileConn)
```

BoW approach

```
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP
```

```
# Reading the transformed file as a dataframe
text <- read.table(file='Sentiment Analysis Dataset.csv', sep=',', header = TRUE)
# Checking the dataframe to confirm everything is in tact
print(dim(text))
```

```
## [1] 100000      2
```

```
# Transforming the text into volatile corpus
train_corp = VCorpus(VectorSource(text$SentimentText))
print(train_corp)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 100000
```

```
# Creating document term matrix
dtm_train <- DocumentTermMatrix(train_corp, control = list( tolower = TRUE,
   removeNumbers = TRUE, stopwords = TRUE, removePunctuation = TRUE, stemming = TRUE ))
# Basic EDA on dtm
inspect(dtm_train)
```

```
## <<DocumentTermMatrix (documents: 100000, terms: 74760)>>
## Non-/sparse entries: 3399444/7472600556
## Sparsity           : 100%
## Maximal term length: 188
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    book get good great just like movi one read time
##   1250     0   0    0     0    0    0    0   0    0    0
##   56817    0   0    0     0    0    0    0   0    0    0
##   63995    0   2    1     1    0    2    0   2    0    1
##   6785     0   7    1     0    0    1    0   1    0    0
##   69262    0   0    0     0    0    0    0   0    0    0
##   73633    1   0    0     2    0    3    0   2    0    2
##   79144    0   0    0     0    0    0    0   0    0    0
##   80872    0   0    0     0    0    0    0   0    0    0
##   85894    0   1    0     0    0    1    0   1    0    0
##   87875    0   0    0     0    0    0    0   0    0    0
```

```
# Removing sparse terms
dtm_train = removeSparseTerms(dtm_train, 0.99)
inspect(dtm_train)
```

```
## <<DocumentTermMatrix (documents: 100000, terms: 645)>>
## Non-/sparse entries: 2131029/62368971
## Sparsity           : 97%
## Maximal term length: 10
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    book get good great just like movi one read time
##   34297    0   1    0     0    2    5    0   3    0    0
##   38984    6   0    0     1    1    0    0   1    0    1
##   42051    3   0    1     0    2    1    0   3    5    1
##   56269    0   1    0     0    1    0    1   1    0    1
##   65117    0   1    1     1    1    0    0   1    0    0
##   65135    0   0    1     2    0    4    0   0    1    0
##   6785     0   7    1     0    0    1    0   1    0    0
##   80366    0   3    1     1    1    1    0   8    0    0
##   87149    6   0    1     0    0    1    0   2    1    0
##   90397    0   2    1     0    3    3    0   2    0    0
```

```
# Splitting the train and test DTM
dtm_train_train <- dtm_train[1:N_t, ]
dtm_train_test <- dtm_train[(N_t+1):N, ]
dtm_train_train_labels <- as.factor(as.character(text[1:N_t, ]$Sentiment))
dtm_train_test_labels <- as.factor(as.character(text[(N_t+1):N, ]$Sentiment))
```

```
# Convert the cell values with a non-zero value to Y, and in case of a zero convert it to N,
cellconvert<- function(x) { x <- ifelse(x > 0, "Y", "N") }
```

```
# Applying the function to rows in training and test datasets
dtm_train_train <- apply(dtm_train_train, MARGIN = 2,cellconvert)
dtm_train_test <- apply(dtm_train_test, MARGIN = 2,cellconvert)
```

Training model

```
# Training the naive bayes classifier on the training dtm
library(e1071)
nb_senti_classifier=naiveBayes(dtm_train_train,dtm_train_train_labels)
# Printing the summary of the model created
summary(nb_senti_classifier)
```

```
##           Length Class  Mode
## apriori     2    table  numeric
## tables    645    -none- list
## levels      2    -none- character
## isnumeric 645    -none- logical
## call        3    -none- call
```

```
# Making predictions on the test data dtm
nb_predicts <- predict(nb_senti_classifier, dtm_train_test,type="class")
```

```
# Computing accuracy of the model
library(rminer)
print(mmetric(nb_predicts, dtm_train_test_labels, c("ACC")))
```

```
## [1] 81.19
```

pretrained word2vec word embedding

```
# Including the required library
library(softmaxreg)
```

```
##
## Attaching package: 'softmaxreg'

## The following object is masked from 'package:e1071':
##
##     sigmoid
```

```
# Importing the word2vec pretrained vector into memory
data(word2vec)
dim(word2vec)
```

```
## [1] 12853    21
```

```r
# Function to get word vector for each review
docVectors = function(x) { wordEmbed(x, word2vec, meanVec = TRUE) }
text = read.csv(file='Sentiment Analysis Dataset.csv', header = TRUE)
# Applying the docVector function on each of the reviews
# Storing the matrix of word vectors as temp
temp=t(sapply(text$SentimentText, docVectors))
dim(temp)
```

```
## [1] 100000     20
```

```r
# Splitting the dataset into train and test
temp_train=temp[1:N_t,]
temp_test=temp[(N_t+1):N,]
labels_train=as.factor(as.character(text[1:N_t,]$Sentiment))
labels_test=as.factor(as.character(text[(N_t+1):N,]$Sentiment))
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.
```

```r
# Training a model using random forest classifier with training dataset
# Observe that we are using 20 trees to create the model
rf_senti_classifier=randomForest(temp_train, labels_train,ntree=20)
print(rf_senti_classifier)
```

```
##
## Call:
##  randomForest(x = temp_train, y = labels_train, ntree = 20)
##                Type of random forest: classification
##                      Number of trees: 20
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 39.82%
## Confusion matrix:
##       1     2 class.error
## 1 23699 15284   0.3920683
## 2 16570 24441   0.4040379
```

```r
# Making predictions on the dataset
rf_predicts<-predict(rf_senti_classifier, temp_test)
library(rminer)
print(mmetric(rf_predicts, labels_test, c("ACC")))
```

```
## [1] 62.875
```

GloVe word embedding

```r
# Including the required library
library(text2vec)
```

```
##
## Attaching package: 'text2vec'

## The following object is masked from 'package:rminer':
##
##     fit
```

```r
# Reading the dataset
text = read.csv(file='Sentiment Analysis Dataset.csv', header = TRUE)
# Subsetting only the review text so as to create Glove word embedding
wiki = as.character(text$SentimentText)
# Create iterator over tokens
tokens = space_tokenizer(wiki)
# Create vocabulary. Terms will be unigrams (simple words).
it = itoken(tokens, progressbar = FALSE)
vocab = create_vocabulary(it)
# Consider a term in the vocabulary if and only if the term has appeared at least
# three times in the dataset
vocab = prune_vocabulary(vocab, term_count_min = 3L)
# Use the filtered vocabulary
vectorizer = vocab_vectorizer(vocab)
# Use window of 5 for context words and create a term co-occurance matrix
tcm = create_tcm(it, vectorizer, skip_grams_window = 5L)
# Create the glove embedding for each in the vocab and
# the dimension of the word embedding should set to 50
# x_max is the maximum number of co-occurrences to use in the weighting function
glove = GlobalVectors$new(rank = 50, x_max = 100)
wv_main = glove$fit_transform(tcm, n_iter = 10, convergence_tol = 0.01)
```

```
## INFO  [14:06:03.185] epoch 1, loss 0.0502
## INFO  [14:06:08.706] epoch 2, loss 0.0317
## INFO  [14:06:14.236] epoch 3, loss 0.0266
## INFO  [14:06:19.757] epoch 4, loss 0.0239
## INFO  [14:06:25.376] epoch 5, loss 0.0221
## INFO  [14:06:30.890] epoch 6, loss 0.0209
## INFO  [14:06:36.293] epoch 7, loss 0.0199
## INFO  [14:06:41.729] epoch 8, loss 0.0191
## INFO  [14:06:47.231] epoch 9, loss 0.0185
## INFO  [14:06:52.696] epoch 10, loss 0.0179
```

```r
# Glove model learns two sets of word vectors - main and context.
# Both matrices may be added to get the combined word vector
wv_context = glove$components
word_vectors = wv_main + t(wv_context)
# Converting the word_vector to a dataframe for visualization
word_vectors = data.frame(word_vectors)
# The word for each embedding is set as row name by default
# Using the tibble library rownames_to_column function, the rownames is copied
```

```r
# as first column of the dataframe
# We also name the first column of the dataframe as words
library(tibble)
word_vectors=rownames_to_column(word_vectors, var = "words")
```

```r
library(softmaxreg)
docVectors = function(x) { wordEmbed(x, word_vectors, meanVec = TRUE) }
# Applying the function docVectors function on the entire reviews dataset
# This will result in word embedding representation of the entire reviews dataset
temp = t(sapply(text$SentimentText, docVectors))
```

```r
# Splitting the dataset into train and test portions
temp_train=temp[1:N_t,]
temp_test=temp[(N_t+1):N,]
labels_train=as.factor(as.character(text[1:N_t,]$Sentiment))
labels_test=as.factor(as.character(text[(N_t+1):N,]$Sentiment))
# Using randomforest to build a model on train data
library(randomForest)
rf_senti_classifier=randomForest(temp_train, labels_train,ntree=20)
print(rf_senti_classifier)
```

```
##
## Call:
##  randomForest(x = temp_train, y = labels_train, ntree = 20)
##                Type of random forest: classification
##                      Number of trees: 20
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 31.77%
## Confusion matrix:
##       1     2 class.error
## 1 26911 12070   0.3096380
## 2 13340 27671   0.3252786
```

```r
# Predicting labels using the randomforest model created
rf_predicts<-predict(rf_senti_classifier, temp_test)
# Estimating the accuracy from the predictions
library(rminer)
print(mmetric(rf_predicts, labels_test, c("ACC")))
```

```
## [1] 71.995
```

fastText word embedding

```r
library(fastTextR)
# Input reviews file
text = readLines("Sentiment Analysis Dataset_ft.txt")
```

```r
# Dividing the reviews into training and test
temp_train=text[1:N_t]
temp_test=text[(N_t+1):N]
```

```r
# Creating txt file for train and test dataset
fileConn<-file("train.ft.txt")
writeLines(temp_train, fileConn)
close(fileConn)
fileConn<-file("test.ft.txt")
writeLines(temp_test, fileConn)
close(fileConn)
# Creating a test file with no labels
temp_test_nolabel<- gsub("__label__1", "", temp_test, perl=TRUE)
temp_test_nolabel<- gsub("__label__2", "", temp_test_nolabel, perl=TRUE)
```

```r
fileConn<-file("test_nolabel.ft.txt")
writeLines(temp_test_nolabel, fileConn)
close(fileConn)
# training a supervised classification model with training dataset file
model<-ft_train("train.ft.txt", method = "supervised", control = ft_control(nthreads = 3L, seed = 1))
# Obtain all the words from a previously trained model
words<-ft_words(model)
```

```r
# Obtain word vectors from a previously trained model.
word_vec<-ft_word_vectors(model, words)
```

```r
# Predicting the labels for the reviews in the no labels test dataset
# Getting the predictions into a dataframe so as to compute performance measurement
ft_preds<-ft_predict(model, newdata = temp_test_nolabel)
# Reading the test file to extract the actual labels
reviewstestfile<- readLines("test.ft.txt")
# Extracting just the labels frm each line
library(stringi)
actlabels<-stri_extract_first(reviewstestfile, regex="\\w+")
# Converting the actual labels and predicted labels into factors
actlabels<-as.factor(as.character(actlabels))
ft_preds<-as.factor(as.character(ft_preds$label))
# Getting the estimate of the accuracy
library(rminer)
print(mmetric(actlabels, ft_preds, c("ACC")))
```

```
## [1] 86.47
```

Drug Data

```r
set.seed(1)
N_Drug = 146942
reviews_text_Drug<-readLines("Drug Train.csv", n = N_Drug)
reviews_text_Drug<-data.frame(reviews_text_Drug)
```

```r
library(tidyr)
reviews_text_Drug<-separate(data = reviews_text_Drug, col = reviews_text_Drug, into = c("Sentiment", "Se
reviews_text_Drug<-reviews_text_Drug[-1,]
N_Drug = N_Drug - 1
```

```r
# Retaining only alphanumeric values in the sentiment column
reviews_text_Drug$Sentiment<-gsub("[^[:alnum:] ]","",reviews_text_Drug$Sentiment)
# Retaining only alphanumeric values in the sentiment text
reviews_text_Drug$SentimentText<-gsub("[^[:alnum:] ]"," ",reviews_text_Drug$SentimentText)
# Replacing multiple spaces in the text with single space
reviews_text_Drug$SentimentText<-gsub("(?<=[\\s])\\s*|^\\s+|\\s+$", "", reviews_text_Drug$SentimentText
```

```r
# Balance our data
minlabel<-names(which(table(reviews_text_Drug$Sentiment)==min(table(reviews_text_Drug$Sentiment))))
maxlabel<-names(which(table(reviews_text_Drug$Sentiment)==max(table(reviews_text_Drug$Sentiment))))

n_maxlabel<-min(table(reviews_text_Drug$Sentiment))
minlabelid<-c(1:N_Drug)[reviews_text_Drug$Sentiment==minlabel]
maxlabelid<-sample(c(1:N_Drug)[reviews_text_Drug$Sentiment==maxlabel],n_maxlabel)
balanceid<-sample(c(minlabelid,maxlabelid))
reviews_text_Drug<-reviews_text_Drug[balanceid,]

N_Drug = nrow(reviews_text_Drug)
N_train_Drug = round(0.8*N_Drug)
```

```r
# Writing the output to a file that can be consumed in other projects
write.table(reviews_text_Drug,file = "Sentiment Analysis Dataset_Drug.csv",row.names = F,col.names = T,s
```

```r
# reading the first 1000 reviews from the dataset
reviews_text_Drug<-readLines("Drug Train.csv", n = 146942)
reviews_text_Drug<-reviews_text_Drug[-1]
reviews_text_Drug<-reviews_text_Drug[balanceid]
# basic EDA to confirm that the data is read correctly
print(class(reviews_text_Drug))
```

```
## [1] "character"
```

```r
print(length(reviews_text_Drug))
```

```
## [1] 80150
```

```r
print(head(reviews_text_Drug,2))
```

```
## [1] "\"2\",\"Clarithromycin Sinusitis Works well very tired after course of clarithyromycin\""
## [2] "\"2\",\"Mesalamine Ulcerative Colitis Active I was diagnosed with ulcerative colitis a year ago
```

```r
# replacing the positive sentiment value 2 with __label__2
reviews_text_Drug<-gsub("\\\"2\\\",","__label__2 ",reviews_text_Drug)
# replacing the negative sentiment value 1 with __label__1
reviews_text_Drug<-gsub("\\\"1\\\",","__label__1 ",reviews_text_Drug)
```

```r
# removing the unnecessary \" characters
reviews_text_Drug<-gsub("\\\"","  ",reviews_text_Drug)
# replacing multiple spaces in the text with single space
reviews_text_Drug<-gsub("(?<=[\\s])\\s*|^\\s+|\\s+$", "", reviews_text_Drug, perl=TRUE)
# Basic EDA post the required processing to confirm input is as desired
print("EDA POST PROCESSING")
```

```
## [1] "EDA POST PROCESSING"
```

```r
print(class(reviews_text_Drug))
```

```
## [1] "character"
```

```r
print(length(reviews_text_Drug))
```

```
## [1] 80150
```

```r
print(head(reviews_text_Drug,2))
```

```
## [1] "__label__2 Clarithromycin Sinusitis Works well very tired after course of clarithyromycin"
## [2] "__label__2 Mesalamine Ulcerative Colitis Active I was diagnosed with ulcerative colitis a year a
```

```r
# writing the revamped file to the directory so we could use it with
# fastText sentiment analyzer project
fileConn<-file("Sentiment Analysis Dataset_ft_Drug.txt")
writeLines(reviews_text_Drug, fileConn)
close(fileConn)
```

BoW approach

```r
# including the required libraries
library(SnowballC)
library(tm)
# setting the working directory where the text reviews dataset is located
# reading the transformed file as a dataframe
text_Drug <- read.table(file='Sentiment Analysis Dataset_Drug.csv', sep=',', header = TRUE)
# checking the dataframe to confirm everything is in tact
print(dim(text_Drug))
```

```
## [1] 80150     2
```

```r
# View(text)
```

```r
# transforming the text into volatile corpus
train_corp_Drug = VCorpus(VectorSource(text_Drug$SentimentText))
print(train_corp_Drug)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 80150
```

```r
# creating document term matrix
dtm_train_Drug <- DocumentTermMatrix(train_corp_Drug, control = list( tolower = TRUE,removeNumbers = TRU
# Basic EDA on dtm
inspect(dtm_train_Drug)
```

```
## <<DocumentTermMatrix (documents: 80150, terms: 44610)>>
## Non-/sparse entries: 2919201/3572572299
## Sparsity           : 100%
## Maximal term length: 95
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    day effect get month pain start take week work year
##   14443   0      7   0     0    0     1    1    0    0    0
##   21739   3      4   3     1   10     4    3    5    6    1
##   32948   9      1   1     1    1     3    5    1    1    5
##   35157   7      7   4     1    3     1    6    4    4    0
##   39889   0      2   2     4    3     4    1    0    2    1
##   4810    0      7   0     0    0     1    1    0    0    0
##   48674   7      7   4     1    3     1    6    4    4    0
##   50714   7      5   3     2    0     4    9    6    2    1
##   56489   0      2   2     4    3     4    1    0    2    1
##   79862   6      0   2     0    7     0    2    0    4    2
```

```r
# Removing sparse terms
dtm_train_Drug = removeSparseTerms(dtm_train_Drug, 0.99)
inspect(dtm_train_Drug)
```

```
## <<DocumentTermMatrix (documents: 80150, terms: 645)>>
## Non-/sparse entries: 2177799/49518951
## Sparsity           : 96%
## Maximal term length: 14
## Weighting          : term frequency (tf)
## Sample             :
##        Terms
## Docs    day effect get month pain start take week work year
##   1194    2      3   4     0    0     1    3    3    1    0
##   21739   3      4   3     1   10     4    3    5    6    1
##   32948   9      1   1     1    1     3    5    1    1    5
##   35157   7      7   4     1    3     1    6    4    4    0
##   35179   2      3   4     0    0     1    3    3    1    0
##   39889   0      2   2     4    3     4    1    0    2    1
##   48674   7      7   4     1    3     1    6    4    4    0
##   50714   7      5   3     2    0     4    9    6    2    1
##   56489   0      2   2     4    3     4    1    0    2    1
##   79862   6      0   2     0    7     0    2    0    4    2
```

```r
# splitting the train and test DTM
dtm_train_train_Drug <- dtm_train_Drug[1:N_train_Drug, ]
dtm_train_test_Drug <- dtm_train_Drug[(N_train_Drug+1):N_Drug, ]
dim(dtm_train_Drug)
```

```
## [1] 80150   645
```

```r
dtm_train_train_Drug_labels <- as.factor(as.character(text_Drug[1:N_train_Drug, ]$Sentiment))
dtm_train_test_Drug_labels <- as.factor(as.character(text_Drug[(N_train_Drug+1):N_Drug, ]$Sentiment))
```

```r
cellconvert<- function(x) { x <- ifelse(x > 0, "Y", "N") }
```

```r
# applying the function to rows in training and test datasets
dtm_train_train_Drug <- apply(dtm_train_train_Drug, MARGIN = 2,cellconvert)
dtm_train_test_Drug <- apply(dtm_train_test_Drug, MARGIN = 2,cellconvert)
# inspecting the train dtm to confirm all is in tact
# View(dtm_train_train)
```

training model

```r
# training the naive bayes classifier on the training dtm
library(e1071)
nb_senti_classifier_Drug=naiveBayes(dtm_train_train_Drug,dtm_train_train_Drug_labels)
# printing the summary of the model created
summary(nb_senti_classifier_Drug)
```

```
##           Length Class  Mode
## apriori     2    table  numeric
## tables    645    -none- list
## levels      2    -none- character
## isnumeric 645    -none- logical
## call        3    -none- call
```

```r
# making predictions on the test data dtm
nb_predicts_Drug <- predict(nb_senti_classifier_Drug, dtm_train_test_Drug,type="class")
# printing the predictions from the model
print(nb_predicts_Drug)
```

```
##     [1] 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 2 1 1 1 1 1 2 1 2 1 2 2 1 1 2 1 2 2 1 2
##    [37] 2 2 2 1 2 2 1 2 2 1 1 1 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 1 2 2 1 1 2
##    [73] 2 2 1 2 2 1 1 2 1 1 2 2 1 2 1 2 2 1 2 2 1 2 2 2 2 1 1 2 2 2 1 2 1 1 1 1
##   [109] 1 1 1 1 1 2 1 2 1 2 2 1 2 2 2 2 2 2 2 1 1 2 1 2 1 2 1 2 2 1 2 1 1 2 2 1
##   [145] 2 2 2 2 1 2 2 1 2 2 1 2 2 1 1 1 2 2 1 2 2 2 2 2 2 2 1 1 1 2 1 2 2 1 1 1
##   [181] 1 2 2 1 1 2 1 2 1 1 1 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2
##   [217] 1 2 2 2 2 1 2 1 1 1 1 2 2 1 1 1 1 2 2 2 2 2 1 2 2 2 2 1 1 2 1 2 1 1 2 2
##   [253] 2 1 1 2 2 2 1 2 1 1 1 2 2 2 2 2 2 2 1 1 2 2 1 1 1 2 1 2 2 2 2 2 1 1 1 1
##   [289] 2 1 2 2 2 1 2 2 2 1 1 2 2 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 1 2 1 1 1
##   [325] 1 2 2 1 2 1 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 1 2 1 1 2 1 2 1 2 1 2 1 1 2
##   [361] 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 1 1 1 2 2 1 2 1 1 1 2 2 2 2 2 2 1 1 2 2 1
```

```
##   [397] 1 2 1 1 2 1 1 1 1 1 1 2 1 2 1 2 2 1 1 2 2 1 2 1 1 2 2 2 1 2 1 2 2 2 2 2
##   [433] 1 1 2 1 2 2 1 1 2 1 1 2 1 2 2 2 2 1 1 2 1 2 1 2 2 1 1 2 2 2 1 2 1 2 1 1
##   [469] 2 2 1 1 2 1 2 1 2 2 1 1 1 2 1 1 1 2 1 2 2 2 2 1 1 2 1 2 2 2 1 2 2 2 2 1
##   [505] 1 2 1 2 1 2 1 2 2 2 1 2 1 2 2 1 2 2 1 1 1 1 1 1 2 2 1 2 2 1 2 1 1 1 1 1
##   [541] 2 1 1 1 1 2 2 1 1 2 1 2 2 2 2 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 2 1 2
##   [577] 2 1 1 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 1 2 1 1 2 1 1 1 2 1 2 2 1 2 1 2 1
##   [613] 2 2 2 2 2 2 1 2 1 1 1 2 1 2 2 2 1 2 1 1 2 2 2 2 1 1 1 2 2 2 1 2 2 2 2 1
##   [649] 1 2 1 1 2 2 2 1 2 1 2 2 1 1 2 1 1 1 2 2 2 2 2 2 2 1 1 2 2 1 1 1 1 1 2 1
##   [685] 1 1 2 2 2 1 2 1 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1 1 2 1 2 1 2 2
##   [721] 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 1 2 1 2 1 2 2 2 2 2 1 2 1 2 2 2 1 2 1
##   [757] 1 1 2 2 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 1 1 2 1 1 2
##   [793] 2 2 2 2 1 2 1 2 2 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 1
##   [829] 1 1 2 2 1 2 2 2 1 2 1 1 2 1 2 2 2 2 2 2 2 2 1 1 2 1 1 1 1 2 1 1 1 2 2 2
##   [865] 2 1 1 1 2 1 2 1 1 2 2 2 2 1 2 2 1 2 1 2 1 2 1 2 1 1 2 1 2 2 1 2 2 2 2 2
##   [901] 2 1 2 1 1 1 1 2 2 2 2 1 2 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 2 1 1 2 2 1 2
##   [937] 1 2 2 2 1 2 1 2 1 2 2 1 2 2 1 2 1 1 1 1 1 1 2 2 1 2 2 1 2 2 2 1 1 2 2 2
##   [973] 1 2 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 1 1 2 2 1 1 1 2 2 2 1 1 1 2 2 1 1 2
##  [1009] 1 2 1 1 2 1 2 1 2 2 1 2 2 2 2 2 2 1 2 2 1 1 1 2 1 2 2 2 1 2 2 2 1 2 2 2
##  [1045] 1 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 1 2 1 2 2 1 1 2 1 1
##  [1081] 2 2 1 1 1 2 2 2 1 2 2 1 1 1 1 1 2 2 2 1 1 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1
##  [1117] 1 1 2 2 2 1 2 1 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 1 2 2 1 2
##  [1153] 1 2 1 2 1 1 2 2 1 2 2 1 2 1 2 1 1 2 2 1 1 2 1 2 1 1 2 1 2 1 1 1 2 2 1 2
##  [1189] 1 1 1 1 2 2 2 1 1 2 1 2 1 2 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 1 1 2 1 2
##  [1225] 1 1 2 2 2 1 1 1 2 2 2 2 2 1 1 2 2 2 1 1 2 1 1 1 2 1 2 2 1 2 2 2 2 1 2 2
##  [1261] 2 2 2 1 1 2 1 2 2 2 2 1 2 2 2 2 1 2 1 1 1 1 1 2 1 2 1 1 1 1 2 1 2 1 1 2
##  [1297] 1 2 1 1 1 2 2 1 2 1 1 1 1 2 2 1 1 1 2 2 1 1 1 2 1 1 2 1 1 2 1 1 2 1 2 1
##  [1333] 1 1 2 2 1 2 1 2 1 1 1 2 1 1 1 2 1 1 2 2 1 2 1 1 2 2 1 1 2 1 2 1 2 2 2 2
##  [1369] 1 2 1 2 2 1 1 2 2 1 2 1 1 2 1 2 1 1 2 2 2 1 2 1 2 2 1 2 2 2 1 1 2 1 1 2
##  [1405] 1 1 1 2 1 1 1 1 2 1 1 1 2 2 1 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 1 2 1 1
##  [1441] 2 2 1 1 2 1 1 1 1 2 2 1 1 2 1 1 2 2 2 1 2 1 2 1 2 1 2 1 2 2 2 2 2 2 1 1
##  [1477] 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 1 2 1 1 2 1 1 2 1 2 1 2 2 1 2 2
##  [1513] 1 1 2 1 1 2 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 2 1 1 1 2 1 1 2 1 2 1
##  [1549] 2 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 2 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 2 2 2
##  [1585] 2 1 2 2 1 1 2 2 2 2 2 1 2 1 1 1 2 1 2 2 2 1 1 2 2 2 2 1 2 2 2 1 1 2 2 1
##  [1621] 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 2 1 2 2 2 2 1 2 1 1 1 2 2 1 1 1 1 2 2 2 2
##  [1657] 2 2 1 2 2 1 1 1 1 1 1 2 2 1 2 2 2 2 2 2 2 1 1 2 1 2 1 2 1 1 2 2 1 1 2 2
##  [1693] 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 1 1 1 2 2 1 1 2 1 1 1 1 2 1 1
##  [1729] 2 1 2 1 1 2 2 2 2 1 2 1 2 2 1 2 1 1 1 1 2 2 2 2 1 2 2 1 1 2 2 2 2 1 1 1
##  [1765] 2 2 1 2 1 2 1 2 2 2 1 1 2 2 2 2 1 2 2 1 2 1 2 2 1 1 1 1 2 2 1 1 2 1 2 1
##  [1801] 1 2 2 2 1 2 2 1 2 1 1 2 1 2 2 2 1 1 2 2 1 2 2 1 1 2 2 1 2 2 1 2 2 1 2 2
##  [1837] 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 1 1 1 1 2 1 1 2 1 2 2 2 1 1 1 2 2 2
##  [1873] 2 1 2 1 1 2 1 2 2 2 1 1 1 2 2 2 2 2 1 2 2 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1
##  [1909] 1 1 1 2 2 1 1 1 2 1 1 2 1 2 2 1 2 2 1 1 2 1 2 1 1 2 2 2 1 1 1 1 2 1 1 2
##  [1945] 2 2 2 2 2 2 2 1 1 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 2 2 2 2 2 1 1 1 1
##  [1981] 2 2 1 1 2 2 2 1 1 1 2 2 1 1 1 2 1 1 2 2 2 1 1 2 1 1 2 1 2 1 2 2 2 2 1 1
##  [2017] 1 1 2 1 2 1 1 2 1 2 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 1 1 2 1 2 2 1
##  [2053] 1 1 1 1 1 2 1 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 1 2 2 1 1 1 1 2 1 2 1 2 2 2
##  [2089] 2 2 1 1 1 1 2 1 2 2 2 1 2 1 1 2 2 2 1 1 1 2 1 2 1 2 1 2 2 2 1 2 2 1 2 2
##  [2125] 1 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 2 1 2 1 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 1
##  [2161] 2 1 2 1 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 2 1 1 2 2 2 2 1
##  [2197] 1 2 2 2 2 1 1 2 2 1 1 1 1 1 2 2 2 2 1 2 2 1 2 2 2 1 1 2 1 1 1 1 1 2 1 2
##  [2233] 1 2 1 1 2 2 2 2 2 2 1 2 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 2 2 2 1 1 2 1
##  [2269] 1 2 2 2 1 1 2 1 1 1 2 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 2 2 2 1 2 2 1 2 2 1
##  [2305] 2 2 1 2 1 2 1 1 1 2 2 2 2 1 1 1 1 2 1 1 2 2 1 1 2 1 2 2 2 2 1 2 1 1 2 1
```

```
##   [2341] 1 1 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 1 1 1 1 2 1 1 2 2 2 2 2 2 1 2 1 2
##   [2377] 2 2 1 1 2 1 1 2 1 1 1 1 2 1 2 1 1 1 1 2 1 2 2 2 1 2 2 2 2 1 1 1 2 1 1 1
##   [2413] 2 1 2 1 2 2 2 2 1 2 2 2 2 1 1 1 2 1 2 1 2 2 1 2 1 1 1 1 2 1 1 2 2 2 2 1
##   [2449] 2 1 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2
##   [2485] 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 1 2 1 2 1 1
##   [2521] 2 1 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 1 2 1
##   [2557] 2 2 2 1 1 2 2 1 1 2 1 1 2 2 1 2 2 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 2 2 1
##   [2593] 1 2 2 1 2 1 2 2 2 2 2 2 1 2 1 1 2 1 2 1 2 2 2 2 2 1 2 2 1 2 2 2 1 2 1 1
##   [2629] 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 1 1 2 1 1 1 2 2 1 1
##   [2665] 1 1 2 1 1 1 2 1 2 2 2 2 1 1 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 2 1 2 2 1 1 1
##   [2701] 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 1 1 1 2 1 2 2 1 1 1 2 2 2 1 1 2 2 2 2 2 2
##   [2737] 2 1 1 2 2 1 1 2 2 2 1 2 2 2 1 1 1 1 2 2 2 2 2 1 2 1 1 1 1 2 1 1 2 1 1 2
##   [2773] 2 2 1 1 1 2 1 1 2 1 2 2 2 1 1 2 1 2 2 1 2 2 2 1 2 2 1 1 2 1 2 1 2 1 1 2
##   [2809] 2 2 2 2 1 2 1 1 2 2 1 2 2 1 1 2 1 2 1 1 2 2 2 1 1 1 2 2 1 2 1 2 2 1 1 1
##   [2845] 1 2 1 1 2 1 2 2 1 2 1 2 1 2 1 1 1 1 2 2 2 1 2 1 2 2 2 2 2 2 1 1 1 1 2 2
##   [2881] 1 2 1 1 1 2 1 2 1 2 1 2 1 2 2 1 1 2 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 1 2 2
##   [2917] 2 1 2 1 2 1 2 2 2 1 1 1 2 1 1 2 1 2 2 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1
##   [2953] 2 2 2 2 2 1 1 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 1 1 1 2 2
##   [2989] 2 2 1 1 2 1 1 1 1 1 2 1 2 2 2 2 2 1 2 1 2 2 2 2 1 1 1 1 1 1 2 1 2 1 1 1
##   [3025] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1 2 1 1 2 2 2 2 1
##   [3061] 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 2 2 2 1 1 2 1 1 2 2 2 1 2 2 2 2 2 1 2 1 1
##   [3097] 1 1 1 1 2 1 2 2 1 2 1 1 2 2 2 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 1 2 1 1
##   [3133] 2 1 1 2 2 1 2 2 1 1 2 2 2 1 2 2 2 2 2 2 1 2 1 1 1 1 1 2 1 2 1 1 2 2 2 1
##   [3169] 1 2 1 1 1 1 2 2 2 1 2 1 2 2 2 2 2 2 1 1 1 1 2 2 2 1 2 1 1 2 1 1 1 2 1 1
##   [3205] 1 2 2 1 1 2 1 1 1 2 1 1 1 2 2 1 1 2 1 1 2 1 1 2 2 1 2 1 2 1 2 2 2 2 1 1
##   [3241] 1 1 1 2 2 1 1 2 2 2 2 1 2 2 2 1 1 2 1 1 1 1 2 1 1 2 1 2 2 1 2 1 1 1 1 1
##   [3277] 1 1 2 1 2 1 2 2 1 1 2 2 1 2 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 1 1 1 1 1 1 1
##   [3313] 1 1 2 2 2 2 2 1 2 1 2 1 1 2 1 2 1 2 2 1 2 2 1 2 2 1 1 1 1 1 1 2 2 1 2 2
##   [3349] 2 2 1 2 2 1 1 2 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 1 1 1 1 2 2 2 1 2 2 1 2
##   [3385] 1 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 1 2 1 2 2 1 2 1 1 2 2 1 1 1 1 2 2 2 1 2
##   [3421] 1 1 1 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2 1 2 2 1 2 2 1 1 1 1 1 1 2 1 1 2
##   [3457] 1 2 2 2 1 1 1 2 2 2 1 2 2 1 2 1 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2 1 2 1 2 2
##   [3493] 2 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 2 1 1 1 2 1 1 2 2 2 2 1 2 1
##   [3529] 1 2 1 1 2 1 2 2 1 1 2 1 2 1 1 2 2 1 2 2 1 1 2 2 1 1 2 1 1 1 2 1 2 1 1 1
##   [3565] 2 2 1 1 2 2 1 1 1 1 1 1 1 2 2 2 2 1 2 2 2 2 2 2 1 1 1 1 2 1 2 2 1 2 1 2
##   [3601] 1 2 2 2 2 1 2 2 2 1 2 1 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 1 1 1
##   [3637] 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 1 2 1 1 2 2 1 2 1 2 2 2 2 1 2 2
##   [3673] 2 1 1 1 2 1 1 2 2 1 1 2 1 2 2 1 2 2 1 2 2 1 2 1 2 2 2 1 1 2 1 2 1 2 1 1
##   [3709] 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 2 1 1 1 2 1
##   [3745] 2 1 2 1 2 1 1 2 2 2 2 2 2 1 2 1 2 1 1 2 1 2 2 2 2 2 1 1 1 2 2 2 1 2 1 1
##   [3781] 1 1 1 2 2 1 1 2 1 2 1 2 2 1 2 1 2 2 2 1 1 2 2 1 1 2 2 2 1 2 2 2 1 2 1 2
##   [3817] 1 1 1 1 1 2 1 1 1 1 2 1 2 2 1 1 1 1 1 1 2 1 1 2 1 2 2 1 1 2 2 2 1 1 2 1
##   [3853] 1 1 1 2 2 1 1 2 2 1 1 2 2 2 1 1 2 2 2 2 2 1 2 1 2 1 2 1 1 1 2 2 2 1 1 2
##   [3889] 1 1 2 1 2 2 2 2 2 1 2 1 2 1 2 2 2 1 2 1 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2
##   [3925] 2 2 1 2 1 1 2 1 1 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 1 1 2 2 2 2 1 2 2 1 2 2
##   [3961] 1 2 2 2 2 2 2 2 1 2 1 1 1 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 1 2 2 1 2
##   [3997] 2 1 2 1 2 2 1 2 2 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 1 1 2 2 2 2 1 1 2 2 2
##   [4033] 1 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 1 1 2 1 2 1 1 1 1 2 2 2 1 1 1 2
##   [4069] 2 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 1 2 2 1 1 2 2 2 1 1 2 2 2 2 1 1 2 2 2 2
##   [4105] 1 2 1 2 1 2 1 2 1 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 1 2 1 2 1 1 2 1 1 1 2
##   [4141] 2 1 1 2 2 2 2 2 1 2 1 1 2 1 2 1 2 2 2 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 2 2
##   [4177] 2 2 2 1 2 1 2 2 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 2 1 1 1 2 1 1 1 2 1 1 2 1
##   [4213] 2 1 2 1 2 2 1 2 2 2 1 1 2 1 1 1 2 1 1 2 2 1 1 2 1 2 1 2 1 2 2 2 1 2 2 1
##   [4249] 2 2 2 1 1 1 1 2 1 2 2 2 2 1 2 1 1 1 1 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2
```

```
## [4285] 2 2 1 1 2 2 1 2 2 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 2 1 2 1 2 2 2
## [4321] 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2 1 1 2 1 1 1 2
## [4357] 2 2 1 2 2 1 1 1 1 1 1 1 1 2 1 2 2 2 1 1 1 2 2 1 2 1 1 1 1 1 1 2 1 1 2 1
## [4393] 2 1 1 1 2 2 2 1 1 2 1 1 1 1 1 2 1 1 1 2 2 1 1 2 1 2 2 2 1 1 2 2 2 1 1 2
## [4429] 1 2 1 1 2 1 2 1 1 1 2 1 1 1 2 1 1 2 1 2 1 1 2 2 2 2 1 1 1 2 1 2 2 1 2 1
## [4465] 2 1 2 2 2 1 2 1 2 2 2 1 2 1 1 1 1 2 2 1 1 2 1 1 1 2 2 1 2 2 2 2 2 2 1 2
## [4501] 2 1 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 1 1 1 2 1 1 1 1
## [4537] 2 2 1 1 1 2 1 1 1 2 1 1 1 1 2 1 2 1 1 2 2 2 1 1 1 1 2 1 2 2 2 1 1 1 2 2
## [4573] 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 1 1 2 1 2 1 1 2 1 1 1 1 2 2 2 2 2 2 2
## [4609] 1 2 1 2 2 2 2 1 1 2 1 2 1 1 1 2 2 2 2 2 2 1 2 2 2 1 2 2 1 2 2 2 1 2 2 1
## [4645] 2 1 1 1 1 1 1 2 2 1 2 1 2 2 1 2 2 2 1 2 2 1 1 1 1 2 1 2 2 1 1 1 1 1 2 1 1
## [4681] 1 1 2 1 1 2 2 2 1 1 1 2 2 2 1 1 2 2 1 1 1 2 2 1 2 2 1 2 1 1 2 2 1 2 2 1
## [4717] 1 2 1 1 1 2 2 1 2 2 2 1 1 1 2 2 1 1 2 1 2 1 2 1 2 2 2 2 2 2 1 2 1 2 2 1
## [4753] 2 2 1 1 2 2 1 1 2 1 1 2 1 1 1 1 2 2 2 2 1 2 1 2 2 1 2 2 1 1 2 1 1 2 2 1
## [4789] 2 2 1 1 2 1 1 1 2 2 2 2 1 1 2 1 2 2 1 1 1 1 2 1 2 1 1 1 2 2 1 1 1 1 1 1
## [4825] 2 2 1 1 1 2 1 2 1 1 2 2 1 1 1 2 1 2 2 2 1 1 2 2 2 1 2 2 2 2 1 1 2 2 1 1
## [4861] 2 2 2 1 2 1 1 2 2 1 2 2 2 1 2 2 1 2 1 1 2 1 1 1 2 2 1 1 1 2 2 1 1 1 1 2
## [4897] 2 1 1 1 2 2 2 1 1 1 2 2 2 1 2 2 1 2 1 1 2 2 2 1 1 2 2 2 1 1 2 2 1 1 2 2
## [4933] 2 2 2 1 2 1 2 2 2 1 2 2 2 2 1 1 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 1 1 2
## [4969] 2 2 1 2 1 2 2 2 2 2 2 2 1 1 2 1 2 1 1 1 1 1 2 1 1 2 2 1 2 1 2 1 2 1 1 2
## [5005] 1 2 2 2 1 2 1 1 2 2 2 1 2 1 2 2 2 1 1 1 2 1 2 2 1 2 1 2 2 1 1 1 2 1 2 2
## [5041] 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 2 2 1 2 2 2 1 1 2 1 2 1 2 2 1 2 1 1 1 1 2
## [5077] 2 1 1 1 2 2 2 2 1 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 1 2
## [5113] 2 2 1 1 2 1 2 2 1 2 1 1 1 1 1 2 2 1 2 2 2 1 2 1 1 1 2 1 2 1 2 2 2 1 2 1
## [5149] 2 1 2 1 1 1 2 1 2 2 2 1 1 1 1 2 2 1 1 1 2 2 2 2 1 1 2 1 2 2 1 2 1 2 2 2
## [5185] 1 2 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 2 2 2 1 2 1 1 1 2 1 1 2 1 2 2 1 1 1 2
## [5221] 1 1 2 2 1 2 2 1 1 2 1 2 2 1 2 1 2 2 2 1 2 1 2 2 2 1 1 1 1 2 2 2 1 2 2 1
## [5257] 2 1 2 2 2 1 2 1 2 2 1 2 2 1 1 2 1 2 1 1 1 1 2 1 2 2 1 1 1 1 1 2 2 2 2 1
## [5293] 1 2 2 2 2 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 1 2 2
## [5329] 2 2 1 1 1 2 1 2 1 2 1 1 2 1 1 2 2 1 1 1 2 2 1 1 2 2 1 1 2 2 2 2 1 2 2 2
## [5365] 2 1 2 1 2 2 2 1 1 2 2 1 2 1 2 1 2 1 2 2 2 1 2 2 1 1 2 1 1 1 2 2 1 2 1 2
## [5401] 2 1 2 2 2 1 2 1 2 2 1 1 2 1 1 2 1 1 2 2 2 1 1 1 2 2 1 2 1 1 1 1 2 1 1 1
## [5437] 2 1 2 2 2 2 2 2 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 1 1 1 1 2 1
## [5473] 1 2 1 2 1 1 1 1 2 2 1 1 1 2 1 2 1 1 1 1 2 1 2 2 1 1 1 1 2 2 2 1 1 2 2 1
## [5509] 2 1 2 1 1 1 1 2 2 2 2 1 2 2 1 1 1 2 2 1 2 1 1 2 1 2 2 1 2 2 1 2 2 1 1 2
## [5545] 2 1 2 1 2 1 1 2 1 2 2 1 2 1 1 1 2 2 1 1 1 2 1 2 1 2 1 2 1 2 2 1 2 1 2 1
## [5581] 1 2 1 1 1 2 1 2 2 1 2 2 1 2 1 2 1 1 1 1 2 1 2 2 1 1 2 2 1 1 2 2 2 1 1 2
## [5617] 2 2 1 1 2 2 1 1 2 2 1 1 1 2 2 2 1 2 2 1 1 1 1 2 1 1 2 2 1 2 1 2 1 1 2 1
## [5653] 2 1 2 1 1 2 1 1 2 2 2 2 2 2 1 1 2 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 2 2 1 2
## [5689] 1 2 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1 2 2 1 1
## [5725] 1 1 2 1 2 1 1 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 1 2 2 1 2 2
## [5761] 2 1 2 1 2 1 1 2 1 2 1 2 2 2 2 2 2 2 1 1 1 2 2 1 1 1 1 2 1 1 1 2 1 2 1 1
## [5797] 2 1 2 2 1 1 2 2 1 2 1 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 1 2 1 1 2 1 2 1 1
## [5833] 1 2 2 2 2 1 2 1 1 2 2 1 1 1 1 1 1 2 1 1 2 2 2 1 1 2 2 2 2 1 1 2 1 2 1 1
## [5869] 2 1 2 1 2 1 2 1 2 1 2 2 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 1 1 1 1 2 1 1 1 2
## [5905] 2 2 2 2 2 1 2 1 1 1 1 2 1 2 1 2 2 1 1 2 1 1 1 2 2 2 1 2 2 1 2 1 1 1 1 1
## [5941] 2 2 2 2 1 2 2 2 1 2 2 1 2 1 2 1 1 1 1 1 2 2 2 2 2 1 2 2 2 1 2 2 1 1 1 1
## [5977] 2 1 2 2 1 2 2 1 2 1 2 1 1 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1
## [6013] 1 1 2 2 1 1 1 1 2 1 2 2 2 2 1 1 2 1 2 1 1 1 2 1 1 1 2 2 2 1 1 1 2 2 1 1
## [6049] 1 2 1 2 1 1 2 2 1 2 2 1 1 2 1 1 2 1 1 1 2 2 1 1 1 1 2 2 1 1 1 2 1 2 1 2
## [6085] 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2 2 1 1 1 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 1 2
## [6121] 1 1 1 2 1 2 1 2 1 2 1 2 2 2 2 2 2 2 1 1 2 2 1 2 1 2 1 2 1 2 2 1 1 1 1 2
## [6157] 2 1 2 2 1 1 2 1 1 2 2 1 2 2 2 1 2 2 2 2 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 1
## [6193] 1 2 2 1 2 1 2 1 1 2 2 1 2 2 1 1 1 2 1 1 2 1 1 2 1 2 1 2 1 2 2 1 1 1 2 2
```

```
##   [6229] 2 2 2 2 2 2 1 2 1 2 1 1 2 2 2 1 2 2 1 2 2 1 1 1 2 1 2 2 1 2 2 2 1 2 1 1
##   [6265] 2 2 2 1 2 1 2 1 1 2 2 1 1 1 2 2 2 1 2 1 1 1 1 2 2 2 2 2 2 1 2 2 1 2 2 2
##   [6301] 1 2 1 1 2 1 2 1 2 1 1 1 1 2 2 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 2 1 1
##   [6337] 2 1 2 1 2 2 1 1 1 1 2 2 2 2 1 1 2 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 2 1 2 2
##   [6373] 1 2 1 1 2 1 2 1 1 2 2 1 1 2 1 2 2 2 1 1 2 2 2 1 2 1 2 1 1 2 2 2 1 1 2 1
##   [6409] 2 2 1 1 1 1 1 1 1 2 1 2 1 2 2 1 2 2 1 1 1 2 1 1 2 1 2 2 2 1 2 2 1 2 1 1
##   [6445] 1 2 1 1 1 2 2 2 2 2 2 1 2 2 1 1 2 2 2 1 2 2 1 1 2 1 1 2 2 1 2 2 1 2 2 1
##   [6481] 1 2 2 2 2 2 1 1 2 1 2 2 2 2 1 1 2 2 1 1 2 1 2 1 2 2 2 2 2 1 1 2 2 1 2 1
##   [6517] 1 1 1 1 2 2 1 2 1 1 2 2 1 2 1 2 1 2 2 2 2 2 2 2 1 1 1 2 1 1 2 2 1 1 2 2
##   [6553] 1 1 2 1 1 2 1 1 1 2 1 2 1 1 2 2 1 1 2 2 1 2 1 1 2 2 2 2 2 2 2 2 1 2 2 2
##   [6589] 2 2 1 2 1 2 2 2 1 2 1 2 1 1 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2
##   [6625] 2 2 2 2 1 1 1 1 1 2 1 2 1 1 1 2 2 1 1 2 1 2 1 1 2 2 2 1 1 2 1 1 1 1 1 2
##   [6661] 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2 1 2 1 1 2 2
##   [6697] 2 1 1 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 1 1 2 1 1 2 1 2 1 1 1
##   [6733] 1 2 2 1 2 1 1 1 2 2 1 2 2 2 2 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 1 1 2 2 2 1
##   [6769] 2 1 2 2 1 2 2 2 1 2 1 1 2 1 2 2 2 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1 2 1
##   [6805] 2 2 1 1 1 2 1 1 2 1 1 1 2 1 2 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 1 1 2 2 1
##   [6841] 2 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 2 1 2 1 2 2 2 1 1 1 2 1 1 2
##   [6877] 2 2 1 1 2 1 1 1 1 2 1 1 1 2 2 1 2 1 2 2 2 1 1 1 2 1 1 1 2 2 2 2 1 1 2 1
##   [6913] 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2 2 1 2 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2
##   [6949] 2 1 1 2 2 2 2 2 1 2 2 1 2 2 1 2 1 1 1 1 2 1 2 1 2 2 2 2 1 1 1 2 1 1 2 1
##   [6985] 1 2 2 2 1 1 1 2 2 2 2 2 2 1 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 1 2 1 2 1 2
##   [7021] 1 2 2 2 2 1 2 2 1 2 2 1 2 1 1 2 2 2 1 1 1 1 2 1 2 2 1 1 2 1 1 2 2 1 1 2
##   [7057] 1 2 2 2 2 1 2 1 1 2 1 2 1 2 2 2 2 2 1 1 2 1 1 1 2 2 2 1 2 1 2 1 2 1 2 1
##   [7093] 1 2 2 2 2 1 2 2 2 2 1 2 2 1 1 1 1 2 2 2 1 1 1 1 2 2 2 2 2 1 2 2 2 2 2 1
##   [7129] 1 2 2 2 2 1 1 2 1 1 2 2 1 1 1 2 1 2 2 1 2 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1
##   [7165] 2 2 1 1 1 1 1 1 1 1 2 2 2 2 1 1 2 1 2 2 1 1 2 2 2 2 2 2 2 1 2 1 2 2 1 1
##   [7201] 1 2 2 2 2 2 2 2 1 1 1 2 2 2 1 1 2 2 2 1 1 2 2 2 1 1 2 2 2 1 2 2 2 1 2 2
##   [7237] 1 1 2 1 2 1 2 2 1 1 1 1 1 2 1 2 2 1 2 1 2 1 2 1 1 1 1 2 1 1 1 2 2 2 1 1
##   [7273] 1 1 2 1 1 1 2 2 2 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 2 1 1 1 2 1 1 1 2
##   [7309] 2 1 1 2 1 2 2 2 1 2 1 2 2 2 1 2 2 1 1 2 1 1 2 1 2 1 1 1 1 2 2 1 1 1 1 2
##   [7345] 1 2 2 2 1 2 1 1 1 2 2 2 2 1 1 2 2 2 1 1 2 1 2 2 2 2 2 1 2 2 2 1 1 2 2 1
##   [7381] 2 2 1 2 1 1 2 2 2 1 1 1 1 1 2 2 2 2 1 2 1 1 1 1 2 2 2 2 1 2 1 1 2 2 1 1
##   [7417] 1 2 2 1 2 2 2 2 1 2 2 1 1 2 2 1 1 2 1 2 2 1 2 2 1 1 1 2 1 2 1 1 2 1 1 1
##   [7453] 2 1 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 1 2 2 2 1 2 1 2 2 2 1 1 1 2 2 2 1 2 2
##   [7489] 1 1 2 1 1 2 2 1 1 2 1 2 1 2 1 1 2 2 1 1 2 2 1 2 2 1 2 1 2 2 1 1 2 2 1 2
##   [7525] 1 2 2 1 2 2 1 2 1 1 2 1 1 2 1 2 2 2 2 2 2 2 1 1 1 1 2 1 2 1 2 1 2 2 1 2
##   [7561] 1 1 2 2 2 2 2 1 2 1 2 1 2 2 1 2 2 2 2 1 2 1 1 2 1 2 1 1 1 1 2 1 2 2 2 1
##   [7597] 1 1 1 2 1 2 2 1 1 2 2 2 1 2 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 2 2 1 2 2 1 1
##   [7633] 2 2 2 2 1 2 1 2 2 2 2 1 2 1 2 1 2 1 2 2 2 2 2 2 1 1 2 1 1 2 2 1 1 1 2 2
##   [7669] 2 2 1 2 1 1 1 2 1 1 1 1 2 1 1 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 2 2 1
##   [7705] 1 1 2 1 1 2 1 2 1 2 1 2 2 2 1 1 1 2 2 1 2 1 2 1 2 1 2 1 2 1 1 1 1 2 1 1
##   [7741] 2 1 2 1 2 2 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 2 1 2 1 2 1 1 2 2 1 1 1 2
##   [7777] 2 1 1 1 2 1 1 2 2 2 1 1 1 2 2 2 2 2 1 2 1 1 1 1 2 2 2 2 1 2 1 1 1 1 2 2
##   [7813] 2 2 1 1 1 2 1 1 1 2 2 1 2 1 2 1 1 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2
##   [7849] 1 1 2 1 1 1 2 2 2 1 1 1 1 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1 1 1 1 1 2 2 2 1
##   [7885] 2 2 1 1 2 2 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 1 1 1 2
##   [7921] 2 2 2 1 1 2 2 1 2 2 2 2 2 1 1 2 1 1 2 1 2 1 2 2 1 1 2 2 1 1 2 2 1 2 1 2
##   [7957] 2 2 1 1 1 1 1 1 2 2 1 2 2 2 2 1 1 1 2 1 1 2 2 1 2 1 1 1 1 1 1 2 1 2 2 1
##   [7993] 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 2 2 1 1 2 2 2 2
##   [8029] 1 2 1 2 2 2 2 2 2 2 1 1 2 2 1 2 1 2 1 2 1 1 2 1 1 2 1 1 2 2 2 2 2 2 2 1
##   [8065] 2 2 1 1 2 1 2 1 1 1 1 2 2 2 1 2 2 2 1 1 1 2 1 2 1 1 2 2 1 2 1 2 2 2 2 1
##   [8101] 1 2 1 2 2 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 1 2 2 2 2
##   [8137] 2 2 1 1 2 1 2 1 1 2 1 2 1 1 1 1 1 2 2 2 2 1 2 2 2 1 2 2 1 1 1 2 1 1 2 1
```

```
##    [8173] 1 1 1 1 1 2 2 1 2 2 1 1 1 2 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 2 2 2 2 2 1
##    [8209] 2 2 2 2 2 2 1 2 2 2 1 2 1 1 1 2 2 1 2 2 2 1 2 1 2 1 1 2 1 2 2 2 1 2 1 2
##    [8245] 2 2 2 1 2 2 1 1 1 2 1 2 1 1 1 2 1 1 2 1 1 2 1 2 1 1 2 1 2 2 2 2 2 2 2 1
##    [8281] 2 1 1 1 1 2 1 2 2 1 2 1 1 2 2 1 2 2 2 1 2 1 1 1 2 2 1 1 2 1 2 2 2 1 2 1
##    [8317] 1 1 2 1 2 2 1 2 2 1 2 1 1 1 1 1 1 2 1 2 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1
##    [8353] 2 1 1 1 2 1 2 2 2 1 1 2 1 1 1 2 2 2 1 1 2 2 2 1 1 2 2 1 1 1 1 2 1 1 2 2
##    [8389] 1 1 2 2 2 1 2 2 1 1 1 1 2 2 2 1 1 2 1 2 1 2 2 2 1 1 2 1 1 2 2 1 2 1 1 1
##    [8425] 2 1 2 1 1 2 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 2 2 2 2 2 1 2 2 1 2 2 1 2
##    [8461] 2 1 2 2 2 2 1 1 1 1 1 2 1 2 2 1 2 2 2 1 2 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1
##    [8497] 2 2 1 1 1 2 2 1 1 2 2 2 2 1 2 1 1 1 2 2 2 1 1 1 2 1 1 2 1 1 1 1 1 2 2 2
##    [8533] 1 1 2 1 1 2 1 2 2 1 2 2 2 2 1 1 1 2 1 2 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 2
##    [8569] 2 1 2 2 1 2 1 1 1 2 1 1 1 2 1 2 1 2 1 2 1 1 1 2 2 2 2 2 1 1 2 2 1 1 2 2
##    [8605] 1 1 1 1 2 2 1 1 1 2 2 1 2 1 2 1 2 2 1 1 1 2 1 2 1 1 2 1 2 2 2 1 1 1 1 1
##    [8641] 1 2 2 2 2 2 2 1 1 2 1 1 1 2 1 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1 1 2 2 1 1
##    [8677] 2 1 2 1 1 2 2 1 2 2 1 2 2 1 2 2 1 1 1 2 2 1 2 2 2 2 1 2 1 1 2 2 1 2 2 1
##    [8713] 2 1 2 2 1 2 1 2 1 1 2 2 2 2 2 2 2 2 1 2 2 1 1 1 1 2 1 2 2 1 2 2 1 1 1 2
##    [8749] 1 1 1 2 1 1 2 2 1 1 1 1 1 1 2 1 2 1 2 2 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1
##    [8785] 2 2 2 1 1 2 1 1 2 2 2 1 1 2 2 2 1 2 1 1 1 2 2 1 1 1 1 2 2 1 1 2 1 1 1 2
##    [8821] 2 2 2 2 1 2 1 2 2 2 1 1 1 2 1 1 2 2 2 1 1 1 1 1 2 1 1 2 2 2 1 1 1 1 1 1
##    [8857] 2 2 1 1 2 1 1 1 1 1 1 1 2 2 2 2 1 2 2 2 2 1 1 1 2 1 2 1 2 1 2 2 2 1 2 2
##    [8893] 2 1 1 1 1 2 2 2 2 1 2 1 1 1 1 1 1 2 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 2
##    [8929] 2 1 2 1 1 1 1 1 1 2 2 1 2 1 2 2 1 2 1 1 2 1 2 2 2 1 1 2 2 1 1 2 1 1 1 1
##    [8965] 1 2 2 2 2 1 1 2 2 2 2 2 2 2 1 1 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 1 1 1 2 2
##    [9001] 1 2 2 2 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 1 1 2 1 2 2 2 1 1 1 2 2 2 2 1 1 2
##    [9037] 2 1 2 2 2 1 1 1 1 2 2 2 1 1 2 2 2 1 1 2 1 1 2 1 2 1 2 2 2 1 2 1 2 2 2 1
##    [9073] 2 2 2 1 1 2 1 1 1 2 1 1 1 2 2 2 1 1 1 2 1 1 2 1 1 2 1 2 1 1 2 2 1 2 1 2
##    [9109] 1 1 1 1 1 2 2 2 2 2 1 1 2 1 2 1 2 2 2 2 2 1 1 2 2 1 1 2 2 1 2 1 2 1 1 1
##    [9145] 1 2 2 2 2 1 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 1 1 2 1 1 1 2 1 2 2
##    [9181] 1 2 1 2 2 2 2 2 1 1 1 1 2 2 1 1 2 1 1 2 2 2 2 1 1 1 1 1 2 2 1 2 1 1 1 2
##    [9217] 1 2 1 2 2 2 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 1 1 1 2 2 1 2 1 1 2 2 2 2 2 2
##    [9253] 2 2 1 2 2 1 2 2 1 2 2 2 2 2 1 2 1 1 2 2 2 2 1 1 2 2 2 1 1 1 1 2 2 1 1 2
##    [9289] 2 2 1 1 2 1 2 1 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 1 1 2 1 1 1 1 2 2 2 2 2 2
##    [9325] 2 2 2 2 2 2 1 2 2 1 2 1 2 1 2 2 1 1 2 1 2 2 2 1 1 1 1 2 1 2 1 1 2 1 1 1
##    [9361] 1 1 1 1 2 1 1 1 2 2 2 1 1 2 1 1 2 2 1 2 1 1 2 2 1 2 1 1 1 1 2 2 1 1 2 1
##    [9397] 2 1 2 2 1 1 1 2 1 2 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 2 2 1 2 2 2 1
##    [9433] 1 2 2 1 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 2 2 1 1 2 2 1 1 2 1 1 2 1 1 2 2 1
##    [9469] 2 2 1 2 2 2 1 1 2 1 2 1 1 2 2 1 2 1 1 2 1 1 2 2 2 2 1 1 2 2 2 2 1 2 2 1
##    [9505] 1 2 1 1 1 2 1 2 1 1 1 2 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 2 1 2 1 2 1 2 1
##    [9541] 2 2 1 1 2 2 1 1 2 1 1 2 2 1 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 1 1 2 2 1 2 1
##    [9577] 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 1 2 2 2 2 1 1 1 1 1 1 1 2 1 1 2 2 2 2
##    [9613] 1 1 2 2 2 2 1 2 2 2 2 2 1 1 2 2 1 1 2 2 2 1 2 2 1 2 1 2 2 2 1 1 2 1 2 1
##    [9649] 1 2 1 2 1 1 1 2 1 2 2 2 2 2 2 1 2 1 2 2 1 2 2 1 2 2 2 1 1 1 1 1 2 1 1 1
##    [9685] 2 2 1 2 1 2 1 2 1 2 2 1 2 1 2 2 1 2 1 1 1 1 2 2 1 2 2 1 2 2 2 1 1 1 1 2
##    [9721] 2 2 2 2 1 2 1 2 1 2 1 1 1 2 2 2 1 1 1 2 1 2 1 1 2 1 1 2 2 2 2 2 2 1 2 2
##    [9757] 1 1 2 2 2 2 1 2 2 1 1 1 1 1 2 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 2 2 1 2
##    [9793] 1 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 2 2 2 1
##    [9829] 1 2 1 1 2 1 1 2 2 1 2 1 1 1 2 2 2 1 2 1 2 1 2 2 1 2 2 1 1 1 2 1 1 2 2 2
##    [9865] 1 2 1 2 1 1 2 2 2 2 1 2 1 1 1 2 1 1 2 2 1 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2
##    [9901] 2 1 2 1 1 2 2 1 2 1 2 2 1 2 1 2 1 1 1 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2 2
##    [9937] 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2 1 1 2 2 1 2 1
##    [9973] 2 1 1 1 1 1 2 2 2 2 1 1 1 2 2 2 2 2 2 1 1 2 1 2 1 2 1 1 2 2 1 2 2 1 2 2
##   [10009] 2 1 2 2 1 2 2 2 2 1 2 1 2 2 1 2 1 2 1 2 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2
##   [10045] 2 2 2 2 2 2 1 2 2 1 2 1 1 2 1 1 2 2 2 2 1 2 2 1 1 1 2 2 2 1 1 1 2 2 1 2
##   [10081] 2 2 2 1 2 2 2 2 1 1 2 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 2 2 2
```

```
## [10117] 1 1 2 2 1 1 1 2 2 1 2 2 1 2 2 1 1 1 2 1 1 2 1 2 1 2 1 2 1 1 1 2 1 1 2 2
## [10153] 2 2 1 2 2 1 1 1 1 2 2 2 1 1 1 1 1 2 1 2 1 2 1 1 1 2 2 2 1 2 2 1 1 2 1 2
## [10189] 1 1 2 1 1 1 1 2 1 1 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 2 2 2 1 2 2 2 2 2 1 1
## [10225] 2 1 2 1 2 2 1 2 2 2 2 1 1 1 1 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 2 1 1 2
## [10261] 1 2 1 2 1 1 2 2 1 2 2 1 1 2 2 1 1 2 2 2 1 2 1 2 2 2 2 2 2 1 2 1 1 1 2 1
## [10297] 1 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 1 2 2 2 1 2 1 2 1 2 2 1 1 2 1 2 2 1 1
## [10333] 2 1 2 2 2 2 2 1 2 1 1 1 2 1 1 1 2 2 1 1 1 1 2 2 1 1 1 2 1 1 1 2 1 2 1 1
## [10369] 2 2 1 1 1 2 2 2 1 1 1 2 1 2 2 2 2 2 1 2 1 1 2 2 1 2 2 1 2 1 2 1 2 1 2 1
## [10405] 1 1 1 1 1 2 1 1 2 2 1 1 1 2 1 1 1 1 2 1 2 1 1 1 2 2 1 1 1 1 2 1 1 2 1 1
## [10441] 2 2 2 2 1 2 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 2 2 1 2 1 1 1 1 1 2 1 1 1
## [10477] 1 2 1 2 2 1 1 1 2 1 2 1 2 2 2 1 1 2 1 2 1 1 1 1 1 2 1 2 2 1 2 2 2 1 2 2
## [10513] 1 1 2 1 2 1 1 1 2 1 2 1 2 1 1 1 2 1 2 2 1 1 2 1 2 2 1 1 2 1 2 1 1 2 2 2
## [10549] 2 2 2 2 2 2 2 1 2 1 1 2 2 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 2 1 2 2 2 2 2 2
## [10585] 2 2 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 2 1 1 2 1 2 1 1 1 2
## [10621] 2 2 1 2 2 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 2 1 2 1 2 2 2 1 1 1 1 2 2 1 2 2
## [10657] 2 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 1 2 2 2 1 2 1 2 1 2 1
## [10693] 2 2 1 2 2 1 2 1 1 2 2 1 1 2 2 1 2 1 2 2 1 2 1 2 2 2 2 2 1 2 1 1 2 1 2 1
## [10729] 2 1 2 2 2 1 1 2 2 1 2 2 2 1 1 1 2 2 1 1 2 2 2 1 1 2 1 2 2 2 2 1 2 2 1 1
## [10765] 2 2 2 2 1 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 1 1 2 2 2 1 1 2 2 2 1 1 1 2 2
## [10801] 2 1 2 1 2 2 1 2 1 2 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 1 2 1 2 2 1 1
## [10837] 2 2 1 2 2 1 2 2 1 2 1 2 1 2 1 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 2 2 2 2 1 1
## [10873] 2 1 1 1 2 2 1 1 2 2 2 2 2 1 1 2 1 1 2 1 1 2 2 1 1 1 2 1 2 2 1 2 2 1 1 2
## [10909] 2 1 2 2 2 2 2 2 1 2 1 2 1 1 2 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 1 2 1 2 2 2
## [10945] 1 2 1 1 2 2 1 1 2 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 2 2 2 1 2 1 1 2 2 2 2
## [10981] 1 2 2 2 2 2 2 2 1 2 1 1 1 1 2 2 2 2 2 1 1 2 2 1 1 2 1 1 2 2 1 2 2 1 1 1
## [11017] 2 2 1 1 1 2 1 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 2 2 2 1 1 2 2 1 2 2 1 2 1
## [11053] 2 1 1 2 2 2 1 1 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 1 1 1 2 2 2 1 2 2 1
## [11089] 1 1 2 2 2 1 2 1 2 1 1 2 2 1 1 1 2 1 1 1 1 2 2 1 2 2 2 2 2 2 1 2 1 2 2 1
## [11125] 2 2 1 1 1 2 1 2 2 2 2 1 2 1 2 1 2 2 2 1 1 1 1 2 2 2 2 1 2 2 1 2 2 2 2 1
## [11161] 1 1 2 2 1 1 1 2 2 2 2 2 1 2 1 1 1 1 2 1 1 2 2 1 1 1 2 2 2 2 1 2 1 1 1 1
## [11197] 2 2 2 2 2 1 1 2 1 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 1 2 2 2 2 1 1 1 2 1 2 2
## [11233] 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 1 2 2 2 1 2 1 2 1 2 2 1 1 1 2 1 1 1 2
## [11269] 1 2 1 2 1 2 1 1 1 1 1 2 1 2 1 2 2 2 1 2 1 2 2 2 2 1 1 2 1 2 1 1 2 1 2 2
## [11305] 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 1 2 2 1 1 1 2 1 1 2 2 2
## [11341] 2 2 1 2 1 1 1 2 2 2 1 1 2 1 1 2 1 1 2 2 2 1 1 2 1 1 1 2 1 2 2 1 2 2 2 2
## [11377] 2 1 1 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 1 1 2 2 1 1 1 2 1 1 2 1
## [11413] 1 2 1 1 2 2 1 2 2 1 2 2 2 1 2 2 1 2 1 2 2 2 2 1 2 1 2 1 2 1 1 1 2 2 2 1
## [11449] 1 2 2 1 2 2 2 1 1 1 2 1 1 2 2 2 1 1 1 1 2 1 1 1 1 1 2 2 1 2 1 1 1 1 1 1
## [11485] 2 2 1 2 1 2 1 1 2 2 2 1 2 1 2 1 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2 1 2
## [11521] 2 2 2 1 1 1 1 1 2 2 2 2 2 1 2 2 1 2 1 1 2 1 1 2 2 2 1 1 1 2 1 1 1 1 2 2
## [11557] 1 2 1 1 1 1 1 1 2 1 1 2 1 1 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 2 2 1 1 1 2 2
## [11593] 2 1 2 2 1 1 2 2 1 1 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 2 2 1 2 1 2 1 1 1 2
## [11629] 2 2 1 1 1 2 2 2 2 2 1 2 1 1 1 2 1 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2
## [11665] 2 2 1 1 2 1 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 2 2 1 1 2 2 1 1
## [11701] 1 1 1 2 1 1 2 2 2 1 1 1 2 1 1 2 2 1 2 2 1 2 1 2 1 1 2 1 1 1 2 1 2 2 1 2
## [11737] 2 1 1 2 2 1 1 1 1 2 1 2 1 2 1 2 2 1 2 2 2 2 1 1 2 2 1 1 2 2 1 1 2 1 2 2
## [11773] 2 1 2 1 2 2 1 2 1 2 2 2 1 2 1 2 1 2 1 1 2 2 1 1 2 2 2 2 2 1 1 2 2 1 2 2
## [11809] 1 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 1 1 1 2 2 1 2 2 1 2 1 2 1 2 1 1 1
## [11845] 1 2 2 1 2 1 1 1 1 2 2 1 1 1 1 2 2 1 2 1 1 1 2 2 1 2 1 1 2 2 2 2 2 2 1 1
## [11881] 1 2 2 2 2 1 1 2 1 1 2 1 1 2 1 2 1 2 2 1 2 1 2 1 1 2 2 2 1 2 1 1 1 1 2 2
## [11917] 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 2 1 2 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 2
## [11953] 1 1 1 2 1 2 2 1 1 1 1 1 2 2 1 1 2 2 2 1 1 2 2 1 2 1 2 1 1 2 2 2 1 1 2 1
## [11989] 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1 2 2
## [12025] 2 2 2 2 2 2 1 2 2 2 2 2 1 1 1 1 2 2 1 2 2 2 2 1 2 2 1 1 2 1 2 1 1 2 2 1
```

```
## [12061] 1 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 2 2 2 1 2 1 1 2 2 1 2 1 1
## [12097] 1 2 1 1 1 2 2 1 1 1 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 1 1
## [12133] 2 1 2 2 2 2 2 1 1 2 1 2 2 2 2 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 1 1 1
## [12169] 1 1 2 1 1 1 2 1 2 1 2 2 2 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1 2 1
## [12205] 2 2 2 1 2 1 2 2 1 1 2 1 2 2 2 2 2 1 1 2 1 1 1 2 1 2 1 1 1 2 2 2 1 2 1 2
## [12241] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 2 2 1 2 2 2 1 1 2 2 2 1
## [12277] 2 2 2 2 1 2 1 1 2 1 2 2 1 1 1 1 2 1 1 2 1 2 2 2 2 2 1 1 2 2 2 1 1 1 1 1
## [12313] 2 1 1 1 2 2 1 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 1 1 1 2 1 1 1 1 2 2 1 2 2 2
## [12349] 2 1 1 2 1 1 1 2 1 2 1 1 2 1 2 2 1 2 2 2 1 1 1 1 1 1 1 2 2 1 1 2 2 2 2 2
## [12385] 1 2 1 1 2 2 1 2 1 1 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 1 1 1 1 1 2 2 2 2
## [12421] 1 2 1 2 2 1 2 2 2 1 2 2 1 2 2 2 2 2 1 2 2 2 1 1 1 2 1 2 1 2 2 2 2 1 1 1
## [12457] 1 2 2 1 1 1 1 1 2 2 2 1 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 2
## [12493] 1 2 2 1 1 2 1 1 1 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 2 1 1 1 2 1 2 1 2 2 2 2
## [12529] 2 1 1 1 1 2 1 2 1 1 1 2 2 2 1 2 2 1 2 1 2 1 2 1 2 1 1 1 2 2 1 1 2 2 1 2
## [12565] 2 2 1 2 1 2 2 2 2 1 1 1 1 1 2 2 1 1 2 1 2 1 2 2 1 1 2 2 2 1 1 2 1 1 2 2
## [12601] 1 1 1 2 2 2 2 2 1 2 1 2 1 2 1 1 1 2 1 1 2 1 1 1 2 2 2 1 2 1 1 1 2 2 2 2
## [12637] 2 1 2 1 2 1 2 2 1 1 1 1 2 2 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 2
## [12673] 1 1 1 1 2 2 1 1 2 2 1 1 2 1 2 2 1 2 1 2 2 2 1 1 2 2 2 1 2 1 2 2 2 1 2
## [12709] 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1 1 2 2 1 2 1 2 1 1 2 2 2 1 2 1 2 2 2 2
## [12745] 2 2 2 2 2 1 1 2 2 2 1 2 2 1 2 2 1 2 2 1 1 2 2 2 2 2 2 1 2 1 1 1 1 1 2 1
## [12781] 1 1 2 1 2 2 1 2 1 1 1 2 2 2 1 1 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 1 2 2 1
## [12817] 2 1 1 2 2 2 1 1 1 1 2 1 1 1 1 2 2 1 1 2 2 2 2 2 1 2 1 2 1 1 1 2 1 1 1 1
## [12853] 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 2 1 2 1 2 1
## [12889] 1 1 1 1 1 2 1 2 2 1 1 2 1 1 1 2 2 2 2 1 1 2 2 2 2 1 1 2 1 1 2 2 1 1 1 2
## [12925] 2 1 1 2 1 1 2 2 1 2 2 1 1 2 2 1 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 1 1 2 2 1
## [12961] 1 1 1 2 2 2 1 2 2 2 2 2 1 1 1 2 2 2 1 2 1 1 1 2 2 1 2 2 2 2 1 2 1 2 2 1
## [12997] 2 1 2 2 2 2 1 1 1 1 2 1 2 2 1 2 1 1 2 2 2 1 1 2 2 2 2 1 2 1 1 1 2 1 2 2
## [13033] 2 1 1 1 2 2 2 1 1 2 2 1 2 2 2 1 1 2 2 1 1 2 1 1 2 2 1 1 2 1 1 2 1 1 1 2
## [13069] 1 2 1 2 2 1 2 2 2 1 2 2 1 2 2 1 2 1 2 1 2 1 1 1 2 2 1 2 1 2 2 1 1 1 2 1
## [13105] 2 2 1 2 1 1 2 1 1 1 1 2 2 1 2 2 1 1 1 1 2 1 1 2 1 2 1 1 2 2 2 2 2 2 2 1
## [13141] 2 1 1 2 1 1 1 1 2 1 2 2 2 1 2 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 2 2 1
## [13177] 2 2 2 1 1 2 1 2 1 2 1 2 2 1 2 1 1 2 2 2 2 1 2 1 1 2 1 2 2 2 1 2 1 1 2 1
## [13213] 2 1 2 1 1 1 2 2 1 1 2 2 2 1 2 1 2 2 1 2 1 2 1 1 1 2 1 2 2 2 2 1 1 2 1 1
## [13249] 1 2 2 2 1 1 2 2 1 2 2 2 1 1 2 2 2 1 1 2 2 1 2 2 2 1 2 1 2 2 1 1 2 2 2 1
## [13285] 2 1 2 2 1 1 1 2 1 2 2 2 2 1 2 1 1 2 1 2 2 2 1 1 1 2 2 2 2 1 1 1 2 2 2 2
## [13321] 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 2 1 2 1 1 2
## [13357] 2 1 1 1 2 1 1 2 2 1 1 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 1 1 1 2 2 2 1 2
## [13393] 1 2 2 1 2 2 1 1 1 1 1 1 1 2 1 2 2 1 1 1 1 2 1 1 2 2 1 1 1 1 1 2 1 1 2 1
## [13429] 1 1 2 1 2 2 2 1 2 2 2 1 1 1 2 1 2 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1
## [13465] 1 1 1 1 2 2 1 2 2 2 1 1 1 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 1 1 2 1 2 2 2
## [13501] 1 2 1 2 1 2 1 1 2 1 1 2 2 2 2 2 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 2 2 1 2 2
## [13537] 1 1 2 1 1 2 2 1 2 1 2 1 2 1 1 1 1 1 1 2 1 2 2 2 2 2 1 1 2 2 2 1 2 2 1 1
## [13573] 1 2 1 1 2 1 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 1 1 1 2 1 2 1 1 2 1 2 2 1 2
## [13609] 1 2 2 2 1 1 2 2 1 1 2 1 2 2 1 2 2 1 1 1 1 2 2 2 1 1 1 1 2 2 2 2 1 2 1 2
## [13645] 2 2 1 1 2 1 1 2 2 1 2 2 2 2 1 2 1 2 1 1 2 1 2 2 2 1 1 2 1 2 2 2 1 2 1 1
## [13681] 2 2 1 2 2 1 2 1 2 2 2 1 2 1 2 1 1 1 2 1 2 1 1 1 1 1 2 2 2 1 2 1 1 2 1 1
## [13717] 2 1 2 2 1 1 2 2 2 1 2 1 1 1 2 1 1 2 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 1 1 1
## [13753] 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 1 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 1
## [13789] 2 2 1 2 1 2 2 1 2 1 1 2 2 2 2 2 2 2 1 1 1 2 1 1 1 2 2 2 1 2 2 1 2 1 2 1
## [13825] 1 2 1 1 2 1 2 2 2 1 2 2 1 1 1 1 1 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2
## [13861] 2 1 1 2 1 1 2 1 2 2 1 2 1 2 2 1 1 1 1 1 2 1 2 1 2 2 2 1 2 2 1 2 2 1 2 1
## [13897] 2 1 1 2 2 1 2 1 1 2 2 1 2 1 1 2 1 2 1 1 1 1 2 1 2 1 2 1 1 1 1 1 2 1 1 2
## [13933] 2 2 2 2 1 1 1 2 1 2 2 1 2 1 2 1 2 2 2 1 2 2 1 1 2 1 1 1 1 1 2 1 2 2 2 2
## [13969] 1 2 2 2 1 1 1 2 2 1 2 2 1 1 1 1 2 1 1 1 2 1 2 2 1 2 1 2 2 2 2 2 2 1 2 2
```

```
## [14005] 1 1 1 1 2 1 2 2 1 2 1 1 2 2 2 1 1 2 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 2 1 2
## [14041] 1 1 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 1 1 2 1 2 2 1 2 1 2 2 1 1 2 1 1 2 2 2
## [14077] 2 2 2 2 1 1 1 2 2 2 2 1 1 1 1 2 1 1 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 2
## [14113] 1 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 1 2 1 1 2 1 1 2 2 1 1
## [14149] 1 1 1 1 2 1 2 2 1 2 1 1 2 2 2 1 1 2 2 1 2 2 2 2 1 1 1 1 1 2 1 2 2 1 1 1
## [14185] 2 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 1 2 1 1 1 2 2 1 2 1 1 1 1 2 2 2 2 2 1
## [14221] 1 1 1 2 1 1 2 1 2 1 1 2 1 2 2 2 2 1 1 2 1 1 1 1 2 2 2 2 2 2 1 2 1 1 2 2
## [14257] 2 2 1 2 1 2 1 2 1 1 1 1 1 1 2 2 2 2 1 1 1 1 2 1 1 2 1 1 1 2 2 1 1 1 1 2
## [14293] 1 1 2 2 2 1 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 2 1 1 2 1 2 2 2 2 1 2 2 2
## [14329] 1 2 1 2 1 1 2 2 2 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 2 2 2 2 2 2 1 2 1 1 1 1
## [14365] 2 2 1 2 2 2 1 2 2 1 1 2 1 2 2 2 1 1 1 1 2 2 1 2 1 2 1 2 1 1 1 2 2 1 2 2
## [14401] 1 1 1 1 2 1 1 1 2 1 2 1 2 2 1 1 1 1 2 1 2 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2
## [14437] 2 2 1 1 2 2 1 2 1 1 1 2 1 1 1 2 2 2 2 2 2 1 2 1 2 1 1 2 1 1 2 2 1 1 1 1
## [14473] 2 2 2 1 2 1 1 1 2 2 2 1 1 1 2 2 2 2 2 2 2 2 1 1 2 2 1 2 1 1 2 2 2 1 1 1 2
## [14509] 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 2 2 1 1 1 2 1 2 2 1 2 2 2 2 1 1 1 2 2 2 1
## [14545] 1 1 1 1 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [14581] 1 2 2 1 1 2 2 2 2 2 1 1 1 2 1 2 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [14617] 1 1 2 1 1 2 1 2 2 2 2 1 2 2 1 1 2 1 2 2 1 1 1 2 1 2 1 2 2 2 1 2 2 2 1 1
## [14653] 2 1 2 1 1 1 1 1 2 2 2 2 1 1 1 2 1 2 2 1 2 1 2 2 1 2 1 1 1 2 2 1 2 2 1 1
## [14689] 1 1 1 1 2 1 2 1 1 2 2 1 1 2 1 2 1 1 1 2 2 2 2 1 2 1 1 1 1 2 1 1 2 2 1 1
## [14725] 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 1 2 1 1 1 1 2 1 2 2 2 2 2 1 1 1 1 2 1 2 1
## [14761] 1 1 1 1 2 1 1 2 2 2 2 2 2 1 2 2 1 2 1 1 2 2 2 2 1 1 1 1 2 2 2 1 1 2 2 2
## [14797] 2 2 2 1 2 2 1 2 1 2 2 1 1 2 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1 2 1 2 2 1 2 2
## [14833] 1 2 2 1 2 2 2 1 2 2 1 2 2 1 2 2 2 1 1 2 1 2 2 2 1 2 1 2 2 2 1 1 2 1 2 1
## [14869] 1 2 2 2 1 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 2 2 2 2 1 1 1 2 1 1 1 2 1 1 1
## [14905] 2 1 2 1 2 1 1 1 2 1 2 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 2 1 2 2 1 2 2 2 2 1
## [14941] 2 1 2 1 2 1 2 1 2 2 1 2 1 1 1 2 1 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 1 2 2
## [14977] 2 2 1 1 2 1 2 1 2 2 2 1 2 1 2 2 2 2 1 1 2 2 1 1 2 2 1 2 2 1 1 2 1 1 1 2
## [15013] 2 2 2 2 2 2 1 2 1 2 1 2 2 1 2 1 2 1 1 2 2 1 2 1 1 2 2 2 1 1 1 2 2 2 1 1
## [15049] 2 1 1 2 1 2 1 2 2 1 2 2 1 1 1 2 1 1 2 1 1 1 2 1 2 2 1 2 2 2 2 2 1 1 1 1
## [15085] 1 1 1 2 2 2 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 2 2 2 2 1 2 2 1 2 1
## [15121] 2 1 2 1 2 2 2 1 2 1 1 2 2 1 2 1 1 1 1 2 1 1 1 1 2 2 2 2 1 2 2 1 1 1 1 2
## [15157] 1 2 1 2 1 1 1 1 1 1 2 1 1 2 2 1 1 2 2 2 1 1 2 1 1 1 1 1 2 2 2 2 1 1 1 2
## [15193] 2 1 2 1 2 1 2 2 2 2 2 1 1 2 1 1 2 2 2 2 1 1 1 1 2 2 1 1 1 1 2 1 1 1 2 2
## [15229] 1 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 1 2 2 1 1 2 1 2 2 2 2 2 2 2 1 1 2 2 2 1
## [15265] 2 1 1 2 2 1 2 1 2 2 1 2 1 2 1 2 1 2 2 1 1 2 2 1 2 2 2 1 2 2 1 1 1 2 1 2
## [15301] 1 1 2 1 2 1 1 2 1 1 1 2 2 2 2 1 2 1 2 1 1 2 1 2 2 1 2 2 2 2 1 2 1 2 2 2
## [15337] 2 2 1 2 2 1 2 2 2 2 1 2 2 1 1 2 1 1 2 1 2 2 2 1 1 2 1 1 2 2 2 1 2 2 2 2
## [15373] 2 2 1 2 1 1 1 2 2 2 1 1 2 2 2 1 2 1 2 1 1 2 2 1 2 2 2 2 2 1 2 1 2 1 1 1
## [15409] 2 2 2 1 1 2 1 2 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 2 2 1 2 1 1 1 2 2
## [15445] 2 2 1 1 2 2 2 1 2 2 1 2 1 1 1 1 2 2 2 2 1 2 2 2 2 1 1 2 1 1 2 2 1 2 2 1
## [15481] 2 2 2 2 1 1 1 1 2 1 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 1 2 2 2 2 1 1 1 2 1
## [15517] 2 2 1 2 1 2 1 2 2 2 1 2 2 1 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 2 2 2 1
## [15553] 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 1 1 2 2 2 1 1 2 1 2 2 2 2 2 1 1 2 1 1 2
## [15589] 1 2 1 2 2 1 1 2 1 2 2 2 2 1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 2 1 2 1 2 2 1
## [15625] 2 2 1 1 2 2 2 1 1 2 2 2 1 2 1 2 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [15661] 2 2 2 2 2 2 2 1 1 2 1 2 1 2 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 1 1 2
## [15697] 1 1 1 1 1 1 2 2 1 1 1 1 1 2 2 2 1 2 2 1 1 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2
## [15733] 2 2 1 2 2 2 1 2 2 2 2 2 1 1 2 1 2 1 1 1 1 2 2 2 2 1 1 1 2 1 1 2 2 2 2 2
## [15769] 2 2 1 2 2 2 1 2 2 1 1 2 2 2 2 1 1 1 1 2 1 1 2 2 1 2 1 1 2 1 1 2 2 2 1 2
## [15805] 1 2 1 1 1 1 2 1 2 1 2 1 1 1 2 2 2 1 1 2 1 1 2 2 1 2 1 2 1 1 1 1 2 2 1 1
## [15841] 2 1 2 1 1 1 1 1 1 2 1 2 2 1 1 2 1 2 1 1 1 1 1 2 1 2 2 1 1 2 2 2 1 1 2 1
## [15877] 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 1 1 1 2
## [15913] 2 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 2 2 1 1 2 1 2 2 2 2 1 1 1 1 2 2 2 2 2
```

```
## [15949] 2 1 1 1 2 2 1 2 1 2 2 1 1 2 1 2 1 2 1 2 2 2 2 1 1 1 2 1 1 2 2 2 1 2 1 2
## [15985] 1 1 2 1 2 1 2 1 1 1 1 2 1 2 2 1 2 2 1 2 2 1 2 1 1 2 1 1 2 2 1 1 1 2 1 2
## [16021] 1 1 2 1 1 2 1 2 2 2
## Levels: 1 2
```

```r
# computing accuracy of the model
library(rminer)
print(mmetric(nb_predicts_Drug, dtm_train_test_Drug_labels, c("ACC")))
```

```
## [1] 74.7723
```

pretrained word2vec word embedding

```r
# including the required library
# install.packages("https://cran.r-project.org/src/contrib/Archive/softmaxreg/softmaxreg_1.2.tar.gz",re
library(softmaxreg)
# importing the word2vec pretrained vector into memory
data(word2vec)
# View(word2vec)
dim(word2vec)
```

```
## [1] 12853    21
```

```r
# function to get word vector for each review
docVectors = function(x) { wordEmbed(x, word2vec, meanVec = TRUE) }
text_Drug = read.csv(file='Sentiment Analysis Dataset_Drug.csv', header = TRUE)
# applying the docVector function on each of the reviews
# storing the matrix of word vectors as temp
temp_Drug=t(sapply(text_Drug$SentimentText, docVectors))
# visualizing the word vectors output
# View(temp)
dim(temp_Drug)
```

```
## [1] 80150    20
```

```r
# splitting the dataset into train and test
temp_train_Drug=temp_Drug[1:N_train_Drug,]
temp_test_Drug=temp_Drug[(N_train_Drug+1):N_Drug,]
labels_train_Drug=as.factor(as.character(text_Drug[1:N_train_Drug,]$Sentiment))
labels_test_Drug=as.factor(as.character(text_Drug[(N_train_Drug+1):N_Drug,]$Sentiment))
# including the random forest library
library(randomForest)
# training a model using random forest classifier with training dataset
# observe that we are using 20 trees to create the model
rf_senti_classifier_Drug=randomForest(temp_train_Drug, labels_train_Drug,ntree=20)
print(rf_senti_classifier_Drug)
```

```
##
## Call:
##  randomForest(x = temp_train_Drug, y = labels_train_Drug, ntree = 20)
```

```
##                Type of random forest: classification
##                      Number of trees: 20
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 31.92%
## Confusion matrix:
##       1     2 class.error
## 1 23000  8985   0.2809129
## 2 11479 20649   0.3572896
```

```
# making predictions on the dataset
rf_predicts_Drug<-predict(rf_senti_classifier_Drug, temp_test_Drug)
library(rminer)
print(mmetric(rf_predicts_Drug, labels_test_Drug, c("ACC")))
```

```
## [1] 71.01684
```

GloVe word embedding

```
# including the required library
library(text2vec)
# reading the dataset
text_Drug = read.csv(file='Sentiment Analysis Dataset_Drug.csv', header = TRUE)
# subsetting only the review text so as to create Glove word embedding
wiki_Drug = as.character(text_Drug$SentimentText)
# Create iterator over tokens
tokens_Drug = space_tokenizer(wiki_Drug)
# Create vocabulary. Terms will be unigrams (simple words).
it_Drug = itoken(tokens_Drug, progressbar = FALSE)
vocab_Drug = create_vocabulary(it_Drug)
# consider a term in the vocabulary if and only if the term has appeared at least three times in the da
vocab_Drug = prune_vocabulary(vocab_Drug, term_count_min = 3L)
# Use the filtered vocabulary
vectorizer_Drug = vocab_vectorizer(vocab_Drug)
# use window of 5 for context words and create a term co-occurance matrix
tcm_Drug = create_tcm(it_Drug, vectorizer_Drug, skip_grams_window = 5L)
# create the glove embedding for each in the vocab and
# the dimension of the word embedding should set to 50
# x_max is the maximum number of co-occurrences to use in the weighting
# function
# note that training the word embedding is time consuming - be patient
glove = GlobalVectors$new(rank = 50, x_max = 100)
wv_main_Drug = glove$fit_transform(tcm_Drug, n_iter = 10, convergence_tol = 0.01)
```

```
## INFO  [14:16:42.664] epoch 1, loss 0.0758
## INFO  [14:16:44.940] epoch 2, loss 0.0489
## INFO  [14:16:47.121] epoch 3, loss 0.0402
## INFO  [14:16:49.328] epoch 4, loss 0.0355
## INFO  [14:16:51.581] epoch 5, loss 0.0324
## INFO  [14:16:53.825] epoch 6, loss 0.0302
## INFO  [14:16:55.996] epoch 7, loss 0.0285
```

```
## INFO  [14:16:58.241] epoch 8, loss 0.0272
## INFO  [14:17:00.432] epoch 9, loss 0.0262
## INFO  [14:17:02.657] epoch 10, loss 0.0253
```

```
# Glove model learns two sets of word vectors - main and context.
# both matrices may be added to get the combined word vector
wv_context = glove$components
word_vectors_Drug = wv_main_Drug + t(wv_context)
# converting the word_vector to a dataframe for visualization
word_vectors_Drug=data.frame(word_vectors_Drug)
# the word for each embedding is set as row name by default
# using the tibble library rownames_to_column function, the rownames is copied as first column of the d
# we also name the first column of the dataframe as words
library(tibble)
word_vectors_Drug=rownames_to_column(word_vectors_Drug, var = "words")
# View(word_vectors)
```

```
library(softmaxreg)
docVectors_Drug = function(x) { wordEmbed(x, word_vectors_Drug, meanVec = TRUE) }
# applying the function docVectors function on the entire reviews dataset
# this will result in word embedding representation of the entire reviews
# dataset
temp_Drug=t(sapply(text_Drug$SentimentText, docVectors_Drug))
# View(temp)
```

```
# splitting the dataset into train and test portions
temp_train_Drug=temp_Drug[1:N_train_Drug,]
temp_test_Drug=temp_Drug[(N_train_Drug+1):N_Drug,]
labels_train_Drug=as.factor(as.character(text_Drug[1:N_train_Drug,]$Sentiment))
labels_test_Drug=as.factor(as.character(text_Drug[(N_train_Drug+1):N_Drug,]$Sentiment))
# using randomforest to build a model on train data library(randomForest)
rf_senti_classifier_Drug=randomForest(temp_train_Drug, labels_train_Drug,ntree=20)
print(rf_senti_classifier_Drug)
```

```
##
## Call:
##  randomForest(x = temp_train_Drug, y = labels_train_Drug, ntree = 20)
##                Type of random forest: classification
##                      Number of trees: 20
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 29%
## Confusion matrix:
##       1     2 class.error
## 1 23384  8600   0.2688844
## 2  9994 22133   0.3110779
```

```
# predicting labels using the randomforest model created
rf_predicts_Drug<-predict(rf_senti_classifier_Drug, temp_test_Drug)
# estimating the accuracy from the predictions
library(rminer)
print(mmetric(rf_predicts_Drug, labels_test_Drug, c("ACC")))
```

```
## [1] 74.82845
```

fastText word embedding

```
# loading the required libary
library(fastTextR)
# reading the input reviews file
# recollect that fastText needs the file in a specific format and we created one compatiable file in
# "Understanding the Amazon Reviews Dataset" section of this chapter
text_Drug = readLines("Sentiment Analysis Dataset_ft_Drug.txt")
# Viewing the text vector for conformation
# View(text)
```

```
# dividing the reviews into training and test
temp_train_Drug=text_Drug[1:N_train_Drug]
temp_test_Drug=text_Drug[(N_train_Drug+1):N_Drug]
# Viewing the train datasets for confirmation
# View(temp_train)
```

```
# creating txt file for train and test dataset
# the fasttext function expects files to be passed for training and testing
fileConn<-file("train_Drug.ft.txt")
writeLines(temp_train_Drug, fileConn)
close(fileConn)
fileConn<-file("test_Drug.ft.txt")
writeLines(temp_test_Drug, fileConn)
close(fileConn)
# creating a test file with no labels
# recollect the original test dataset has labels in it
# as the dataset is just a subset obtained from full dataset
temp_test_Drug_nolabel<- gsub("__label__1", "", temp_test_Drug, perl=TRUE)
temp_test_Drug_nolabel<- gsub("__label__2", "", temp_test_Drug_nolabel, perl=TRUE)
# View(temp_test_nolabel)
```

```
fileConn<-file("test_Drug_nolabel.ft.txt")
writeLines(temp_test_Drug_nolabel, fileConn)
close(fileConn)
# training a supervised classification model with training dataset file
model_Drug<-ft_train("train_Drug.ft.txt", method = "supervised", control = ft_control(nthreads = 3L, se
# Obtain all the words from a previously trained model=
words_Drug<-ft_words(model_Drug)
# viewing the words for confirmation. These are the set of words present in our training data
# View(words)
```

```
# Obtain word vectors from a previously trained model.
word_vec_Drug<-ft_word_vectors(model_Drug, words_Drug)
# Viewing the word vectors for each word in our training dataset
# observe that the word embedding dimension is 5
# View(word_vec)
```

```
# predicting the labels for the reviews in the no labels test dataset
# getting the predictions into a dataframe so as to compute performance measurement
ft_preds_Drug<-ft_predict(model_Drug, newdata = temp_test_Drug_nolabel)
# reading the test file to extract the actual labels
```

```r
reviewstestfile_Drug<- readLines("test_Drug.ft.txt")
# extracting just the labels frm each line
library(stringi)
actlabels_Drug<-stri_extract_first(reviewstestfile_Drug, regex="\\w+")
# converting the actual labels and predicted labels into factors
actlabels_Drug<-as.factor(as.character(actlabels_Drug))
ft_preds_Drug<-as.factor(as.character(ft_preds_Drug$label))
# getting the estimate of the accuracy
library(rminer)
print(mmetric(actlabels_Drug, ft_preds_Drug, c("ACC")))
```

```
## [1] 78.67748
```