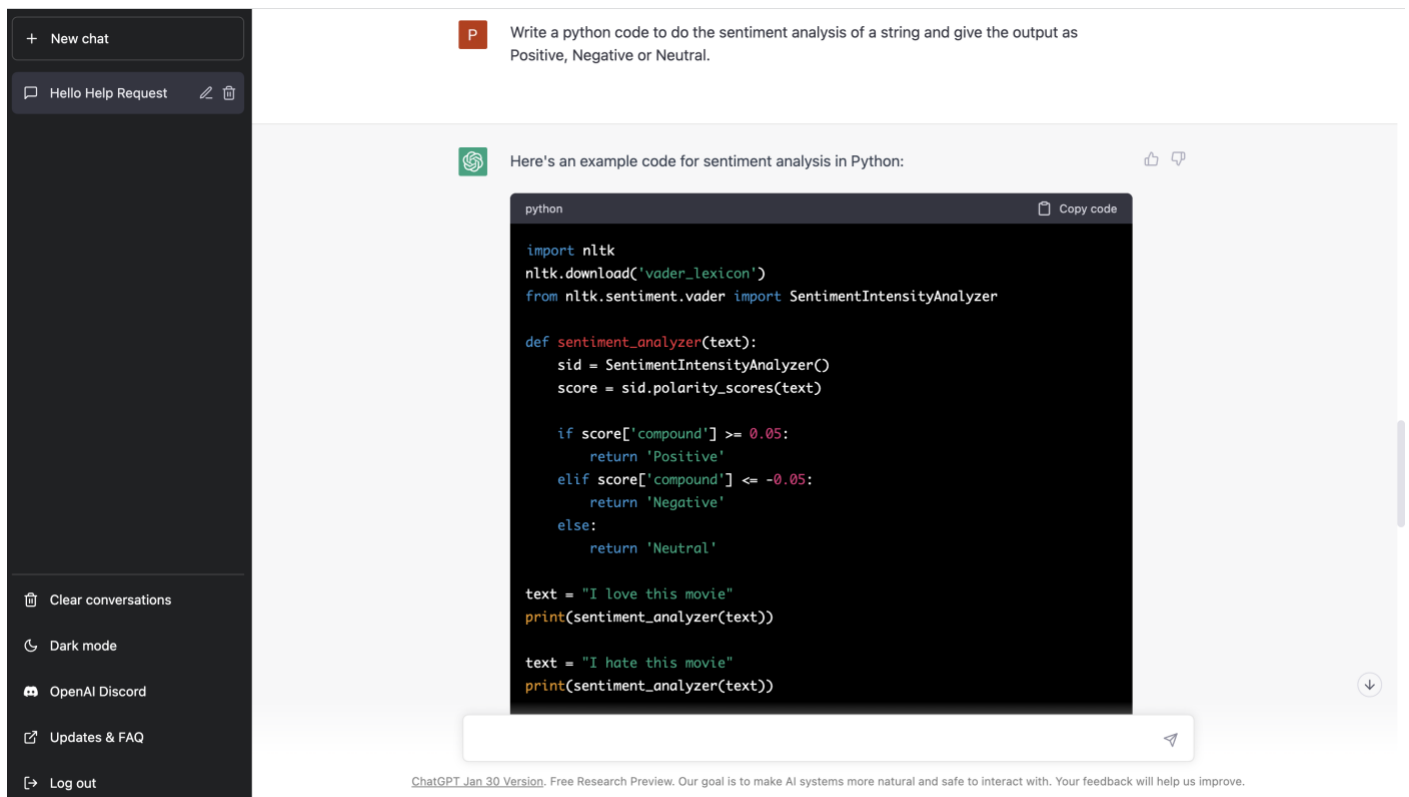


Data 228

IN-CLASS EXERCISE: 1

Group Name: Data Divers

1. Sentiment Analysis: Below are the screenshots of the program generated by ChatGPT



The screenshot displays the ChatGPT web interface. On the left is a sidebar with a 'New chat' button, a chat history entry 'Hello Help Request', and a list of settings: 'Clear conversations', 'Dark mode', 'OpenAI Discord', 'Updates & FAQ', and 'Log out'. The main chat area shows a user prompt: 'Write a python code to do the sentiment analysis of a string and give the output as Positive, Negative or Neutral.' The AI response, starting with the OpenAI logo, says 'Here's an example code for sentiment analysis in Python:' and includes a code block. The code block is titled 'python' and has a 'Copy code' button. The code defines a function 'sentiment_analyzer' that uses 'SentimentIntensityAnalyzer' from 'nltk.sentiment.vader' to analyze text. It returns 'Positive', 'Negative', or 'Neutral' based on the compound score. Two example calls are shown: 'I love this movie' and 'I hate this movie'. At the bottom of the interface, a footer reads: 'ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.'

```
python
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

def sentiment_analyzer(text):
    sid = SentimentIntensityAnalyzer()
    score = sid.polarity_scores(text)

    if score['compound'] >= 0.05:
        return 'Positive'
    elif score['compound'] <= -0.05:
        return 'Negative'
    else:
        return 'Neutral'

text = "I love this movie"
print(sentiment_analyzer(text))

text = "I hate this movie"
print(sentiment_analyzer(text))
```

+ New chat

Hello Help Request

Clear conversations

Dark mode

OpenAI Discord

Updates & FAQ

Log out

optimize the above code

Here's an optimized version of the sentiment analysis code:

```
python
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sentiment_analyzer = SentimentIntensityAnalyzer().polarity_scores

def get_sentiment(text):
    score = sentiment_analyzer(text)

    if score['compound'] >= 0.05:
        return 'Positive'
    elif score['compound'] <= -0.05:
        return 'Negative'
    else:
        return 'Neutral'

text = "I love this movie"
print(get_sentiment(text))

text = "I hate this movie"
print(get_sentiment(text))
```

ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

+ New chat

Hello Help Request

Clear conversations

Dark mode

OpenAI Discord

Updates & FAQ

Log out

it's more descriptive and easier to understand what the function does. The rest of the code remains the same, with the only change being the reference to `sentiment_analyzer` instead of `SentimentIntensityAnalyzer().polarity_scores`.

```
P
Give me the output for
text = "I love this movie"
print(get_sentiment(text))

text = "I hate this movie"
print(get_sentiment(text))

text = "I feel neutral about this movie"
print(get_sentiment(text))
```

The output of the sentiment analysis for the three given sentences would be:

```
Positive
Negative
Neutral
```

ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

2. Use a language model to help your team solve HackerRank coding problems, in the language of your choice

1. Easy:

The screenshot shows the ChatGPT web interface at chat.openai.com. The left sidebar contains navigation options: New chat, Hello Help Request, Clear conversations, Dark mode, OpenAI Discord, Updates & FAQ, and Log out. The main chat area displays a conversation where the user asks for a Python code to solve a problem. The AI responds with a sample Python code that implements the conditions described. The code is as follows:

```
python
def weird_or_not(n):
    if n % 2 != 0:
        print("Weird")
    elif 2 <= n <= 5:
        print("Not Weird")
    elif 6 <= n <= 20:
        print("Weird")
    else:
        print("Not Weird")

num = int(input())
weird_or_not(num)
```

Below the code, the AI explains the logic: "This code first checks if the number is odd by using the modulus operator ('%') to see if the remainder of dividing the number by 2 is 0. If the number is odd, the code prints 'Weird'." The interface also shows a 'Copy code' button and a 'Print Weird if the number is weird. Otherwise, print Not Weird.' instruction.

The screenshot shows the HackerRank 'Weird or Not Weird' challenge page. The page is divided into several sections:

- Task:** Given an integer, n , perform the following conditional actions:
 - If n is odd, print `Weird`
 - If n is even and in the inclusive range of 2 to 5, print `Not Weird`
 - If n is even and in the inclusive range of 6 to 20, print `Weird`
 - If n is even and greater than 20, print `Not Weird`
- Input Format:** A single line containing a positive integer, n .
- Constraints:**
 - $1 \leq n \leq 100$
- Output Format:** Print `Weird` if the number is weird. Otherwise, print `Not Weird`.
- Sample Input 0:** 3
- Sample Output 0:** Weird
- Explanation 0:** $n = 3$. n is odd and odd numbers are weird, so print `Weird`.

The right side of the page shows the submission interface. It includes a 'Run Code' button, a 'Submit Code' button, and a 'Test against custom input' checkbox. A notification states: "You have earned 10.00 points! You are now 25 points away from the 1st star for your python badge." Below this, a 'Congratulations' banner says: "You solved this challenge. Would you like to challenge your friends?" with social media icons and a 'Next Challenge' button. The bottom section shows the test cases and the compiler message:

Test case	Compiler Message
Test case 0	Success
Test case 1	
Test case 2	Input (stdin)
Test case 3	Expected Output
Test case 4	
Test case 5	
Test case 6	

2. Medium:

HackerRank

Prepare > Python > Itertools > Iterables and Iterators

Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

The `itertools` module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an iterator algebra making it possible to construct specialized tools succinctly and efficiently in pure Python.

To read more about the functions in this module, check out their [documentation here](#).

You are given a list of N lowercase English letters. For a given integer K , you can select any K indices (assume 1-based indexing) with a uniform probability from the list.

Find the probability that at least one of the K indices selected will contain the letter: 'a'.

Input Format

The input consists of three lines. The first line contains the integer N , denoting the length of the list. The next line consists of N space-separated lowercase English letters, denoting the elements of the list.

The third and the last line of input contains the integer K , denoting the number of indices to be selected.

Output Format

Output a single line consisting of the probability that at least one of the K indices selected contains the letter: 'a'.

Note: The answer must be correct up to 3 decimal places.

Constraints

$1 \leq N \leq 10$
 $1 \leq K \leq N$

All the letters in the list are lowercase English letters.

Change Theme

Language Python 3

⌕

⋮

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2
3 import math
4
5 def at_least_one_a(n, l, k):
6     num_a = l.count("a")
7     probability = 1.0 - math.comb(n - num_a, k) / math.comb(n, k)
8     return probability
9
10 n = int(input().strip())
11 l = input().strip().split()
12 k = int(input().strip())
13
14 print("{:.3f}".format(at_least_one_a(n, l, k)))
15
16
17
```

Line: 15 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

HackerRank

Prepare > Python > Itertools > Iterables and Iterators

Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

The `itertools` module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an iterator algebra making it possible to construct specialized tools succinctly and efficiently in pure Python.

To read more about the functions in this module, check out their [documentation here](#).

You are given a list of N lowercase English letters. For a given integer K , you can select any K indices (assume 1-based indexing) with a uniform probability from the list.

Find the probability that at least one of the K indices selected will contain the letter: 'a'.

Input Format

The input consists of three lines. The first line contains the integer N , denoting the length of the list. The next line consists of N space-separated lowercase English letters, denoting the elements of the list.

The third and the last line of input contains the integer K , denoting the number of indices to be selected.

Output Format

Output a single line consisting of the probability that at least one of the K indices selected contains the letter: 'a'.

Note: The answer must be correct up to 3 decimal places.

Constraints

$1 \leq N \leq 10$
 $1 \leq K \leq N$


All the letters in the list are lowercase English letters.

Upload Code as File

Test against custom input

Run Code

Submit Code

 You have earned 40,00 points!
You are now 25 points away from the 4th star for your python badge.

77%

195/220

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Input (stdin)

Download

Expected Output

Download

```
1 4
2 a a c d
3 2

1 0.833333333333
```

3. Hard:

Problem

Submissions

Leaderboard

Discussions

You are given a function $f(X) = X^2$. You are also given K lists. The i^{th} list consists of N_i elements.

You have to pick one element from each list so that the value from the equation below is maximized:

$$S = (f(X_1) + f(X_2) + \dots + f(X_k)) \% M$$

X_i denotes the element picked from the i^{th} list. Find the maximized value S_{max} obtained.

$\%$ denotes the modulo operator.

Note that you need to take exactly one element from each list, not necessarily the largest element. You add the squares of the chosen elements and perform the modulo operation. The maximum value that you can obtain, will be the answer to the problem.

Input Format

The first line contains 2 space separated integers K and M .

The next K lines each contains an integer N_i , denoting the number of elements in the i^{th} list, followed by N_i space separated integers denoting the elements in the list.

Constraints

- $1 \leq K \leq 7$
- $1 \leq M \leq 1000$
- $1 \leq N_i \leq 7$
- $1 \leq \text{Magnitude of elements in list} \leq 10^9$

Output Format

Output a single integer denoting the value S_{max} .

Change Theme

Language Python 3

Exit Full Screen View

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 def maxValue(K, M, arr):
3     from itertools import product
4     max_value = 0
5     for combination in product(*arr):
6         value = sum(i**2 for i in combination) % M
7         max_value = max(max_value, value)
8     return max_value
9
10 K, M = map(int, input().strip().split())
11 arr = [list(map(int, input().strip().split()[1:])) for _ in range(K)]
12 print(maxValue(K, M, arr))
13
```

Line: 13 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

You are given a function $f(X) = X^2$. You are also given K lists. The i^{th} list consists of N_i elements.

You have to pick one element from each list so that the value from the equation below is maximized:

$$S = (f(X_1) + f(X_2) + \dots + f(X_k)) \% M$$

X_i denotes the element picked from the i^{th} list. Find the maximized value S_{max} obtained.

$\%$ denotes the modulo operator.

Note that you need to take exactly one element from each list, not necessarily the largest element. You add the squares of the chosen elements and perform the modulo operation. The maximum value that you can obtain, will be the answer to the problem.

Input Format

The first line contains 2 space separated integers K and M .

The next K lines each contains an integer N_i , denoting the number of elements in the i^{th} list, followed by N_i space separated integers denoting the elements in the list.

Constraints

- $1 \leq K \leq 7$
- $1 \leq M \leq 1000$
- $1 \leq N_i \leq 7$
- $1 \leq \text{Magnitude of elements in list} \leq 10^9$

Output Format

Output a single integer denoting the value S_{max} .

Upload Code as File

Test against custom input

Run Code

Submit Code

Python

You have earned 50.00 points!

You are now 65 points away from the 4th star for your python badge.

41%

155/200

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hacks.

Unlock