

JavaScript Most Common Word

Challenge

Given a string `paragraph` and a string array of the banned words `banned`, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique.

The words in `paragraph` are case-insensitive and the answer should be returned in lowercase.

1st Example

```
Input: paragraph = 'Bob hit a ball, the hit BALL flew far after it was hit.',  
       banned    = ['hit']
```

Output: 'ball'

Explanation:

'hit' occurs 3 times, but it is a banned word. 'ball' occurs twice (and no other word does), so it is the most frequent non-banned word in the paragraph. Note that words in the paragraph are not case sensitive, that punctuation is ignored (even if adjacent to words, such as 'ball,'), and that 'hit' isn't the answer even though it occurs more because it is banned.

2nd Example

Input: paragraph = 'a.', banned = []

Output: 'a'



Constraints

- `1 <= paragraph.length <= 1000`
- `0 <= banned.length <= 100`
- `1 <= banned[i].length <= 10`
- Paragraph consists of English letters, space ' ', or one of the symbols: '!', '?', ';', '.'.
- `banned[i]` consists of only lowercase English letters.

Solution

```
const mostCommonWord = (paragraph, banned) => {  
  const bannedSet = new Set(banned),  
        words      = paragraph.toLowerCase().split(/\W+/),  
        map         = {};  
  let answer = {count: 0, word: ''};  
  
  for (const w of words) {  
    if (!bannedSet.has(w)) {  
      if (map[w] == null) {  
        map[w] = 0;  
      }  
  
      map[w]++;  
    }  
  }  
}
```



Solution continues on next page...

```
        if (map[w] > answer.count) {  
            answer.count = map[w];  
  
            answer.word = w;  
        }  
    }  
}  
  
return answer.word;  
};
```

Explanation

I've built a function called `mostCommonWord` that takes in a paragraph and an array of banned words as parameters. Its purpose is to find and return the most common word in the paragraph that is not in the banned words list.

Inside the function, a `bannedSet` variable is created using the `Set` constructor, which converts the `banned` array into a set for efficient lookup.

The `paragraph` is then converted to lowercase and split into an array of words using the `split` method with a regular expression `\W+`, which matches one or more non-word characters (such as punctuation or whitespace).

An empty `map` object is created to store the frequency of each word encountered in the paragraph.

An `answer` object is initialized with initial values of `count: 0` and

`word: ''`. This object will be used to store the most common word and its frequency.

A `for...of` loop is used to iterate over each word in the `words` array.

Inside the loop, it checks if the current word is not in the `bannedSet` using the `has` method. If it is not banned, it proceeds.

Then it checks if the `map` object does not have the current word as a key. If the word is not already in the map, it initializes it with a value of `0`.

It increments the value of the current word in the `map` object to keep track of its frequency.

Next, it checks if the frequency of the current word is greater than the current `answer.count`. If it is, it updates the `answer.count` and `answer.word` with the current word, representing the new most common word.

After the loop finishes, the function returns the `answer.word`, which represents the most common word in the paragraph that is not in the banned words list.

In summary, the `mostCommonWord` function finds the most common word in a paragraph while excluding any banned words. It uses a set for efficient banned word lookup and a map to keep track of word frequencies.