Netflix Cracker Sheet

Note: [To Use this sheet optimally go and watch the "Cracking the Netflix coding interview🔥: The definitive prep guide" video on Debug Buzz Channel.
Link - https://youtu.be/3pc_ieOZ4hM]

1)

Given two strings a and b we need to add ith elements from both. Time limit 0.5secs (AFAIR). (Input size limits I do not remember)

Eg1:

a = "99"

b = "99"

ans = "1818"

Eg2:

a = "9"

b = "11"

ans = "110"

2)

Given an array of integers and an integer K, find the number of subarrays with at least K distinct integers.

Example 1:

array = [1, 2, 1, 1] and k = 2

output=2; {1,2} and {2,1} are only possibilities

Example 2:

array = [1, 2, 3, 4, 1] and k = 4

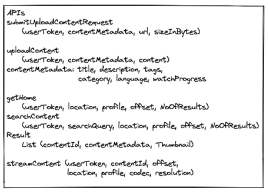
output=6; {1, 2, 3}, {1, 2, 3, 4}, {1,2,3,4,1}, {2,3,4}, {2,3,4,1} and {3,4,1} are the only possibilities. Note that {1,2,3,4,1} is valid because it still has 3 distinct integers.

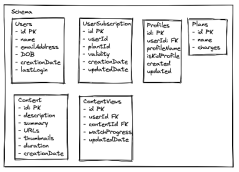
3)

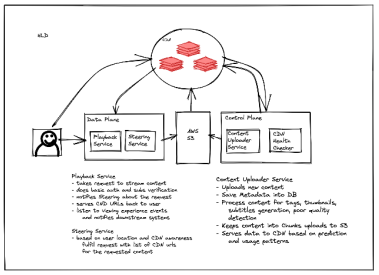
design a system that counts how minutes watched on particular video, number of video watched completely and Category of videos most watched.

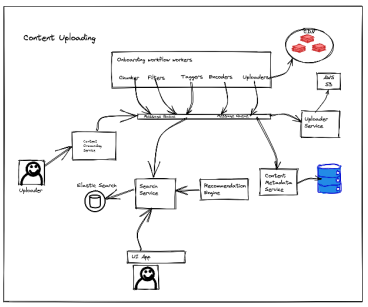
4)

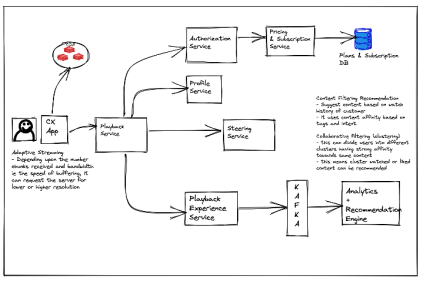
Netflix System Design -



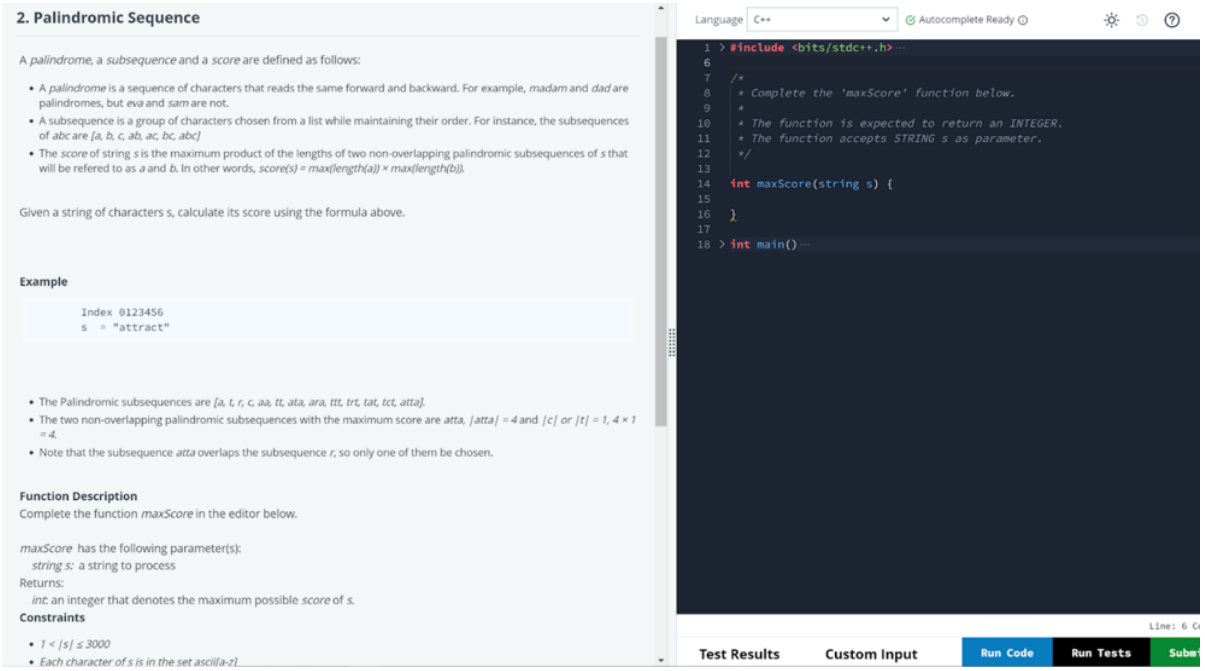








5)



Returns:

int: an integer that denotes the maximum possible *score* of *s*.

Constraints

- $1 < |s| \leq 3000$
- *Each character of s is in the set ascii[a-z]*

▶ Input Format For Custom Testing

▼ Sample Case 0

Sample Input 0

```
STDIN                Function
Parameters
-----                ----------
---------
acdapmpomp →         s =
"acdapmpomp"
```

Sample Output 0

```
15
```

Explanation 0

- The Palindromic subsequences are *[a, c, d, p, m, o, aa, aca, ada, pmp, pmpmp]*.
- The two non-overlapping palindromic subsequences with the maximum score are *pmpmp*, $|pmpmp| = 5$ and $|ada|$ *or* $|aca| = 3$. The maximal score $= 5 \times 3 = 15$.

▶ Sample Case 1

6)

Netflix and chill is a pretty common phrase these days. How about you share your ideas about how you'd design it if asked to do so?

Some questions to clear ambiguity :

- **Q. What will be the main feature of this platform?**
  A. Basically it'll recommend you videos at any time of the day.
- **Q. Who are the targeted audience?**
  A. People all around the world.
- **Q. How will audience access it?**
  A. Using devices like phones, laptops, TVs, etc.

Functional requirements

- **Users can watch videos on the platform**
- **Like/ dislike option**
- **View Count**

Non-functional requirements

- **The platform should be highly available**
- **The response time for users in different regions should be at the same level**
- **The platform should scale while userbase is increasing**

7)

Find how many people are streaming a tv show on netflix simultaneously

8)

Reorder a linked list with all the odd indexed nodes preceeding the evens.

9)

Let's say you have an array of similar json objects. Example object:

```
{
  "field1": "bar",
  "field2": 1,
  "field3": true,
  "field4": [1, 2, 3],
  "field5": {
    "nested": {
            "other": [4, 5]
      }
  }
  ...
}
```

Design a method to query this array. You have to design the code and the format of the query:

```
def search(docs, query):
  pass
```

Query examples:

- **Get documents where field 1 is equal to "bar"**
- **Get documents where field 4 contains (or doesn't contain) 2**
- **Get documents where field5.nested.other contains 4**
- **Get docs where field3 is True**
- **etc.**

10)

design an anonymous web app like Reddit.

- **Users can upvote**
- **Users can downvote**
- **How would you combact spam?**
- **Extended feature: What if all of a sudden we want to turn this app into a social media network (i.e. no longer anonymous)**

11)

Imagine that you have an infinite sorted list of words. Assuming that your word of interest (targeted_word) is at the K position (K is very large). You start at the position x (x less than k). If word[x] < targeted_word, then you move x/2 position, i.e. x = x + x / 2. What is the complexity in term of K. If it is in a log form, then what is the base of the log.

12)

Design a distributed database that syncs across 3 regions and 3 zones within the regions.

Requirement: eventually consistent system

13)

How would you design netflix recommendation engine? Like movie suggestion based on history and interest?