

Concatenación de lenguajes

Objetivo: Resolución del problema numero 10887 del archivo de problemas online del IACM-ICPC para que consiga superar los test de UVA Online Judge, siendo así considerado válido por la organización que provee el problema. Como objetivo secundario se intentará quedar en el top 20 de mejores tiempos de ejecución de dicho problema.

Introducción

La información del problema proporcionada por la IACM-ICPC describe lo siguiente: Un lenguaje es un conjunto de cadenas. Y la concatenación de dos lenguajes es el conjunto resultante de concatenar las cadenas del segundo lenguaje al final de las cadenas del primer lenguaje.

Por ejemplo, si tenemos dos lenguajes A y B tales que:

A = {cat, dog, mouse}

B = {rat, bat}

La concatenación de A y B será:

C = {catrat, catbat, dograt, dogbat, mouserat, mousebat}

Dados dos lenguajes, tu tarea será solamente contar el número de cadenas en la concatenación de dichos lenguajes.

Input

Pueden haber muchos casos de pruebas. La primera línea de la entrada debe contener el número de casos de prueba, T ($1 \leq T \leq 25$). Seguidos, los T casos. La primera línea de cada caso debe contener dos enteros, M y N ($M, N < 1500$), El número de cadenas de cada lenguaje. Entonces las siguientes M líneas contienen las cadenas del primer lenguaje. Las otras N líneas siguientes te dan el segundo lenguaje. Puedes asumir que esas cadenas están formadas por letras minúsculas ('a' hasta la 'z') solo, y de una longitud menor a 10 caracteres y estarán presentadas en una línea separada sin ningún espacio alrededor. Las cadenas de la entrada pueden no estar ordenadas y no estarán duplicadas.

Ejemplo Input:

2

3 2

Cat

Dog

Mouse

Rat

Bat

1 1

abc

cab

Output

Para cada uno de los casos de prueba se deberá mostrar solo una línea de output. El output de cada test comenzará con el número serial de dicho test, seguido por el número de cadenas en la concatenación del segundo lenguaje tras el primero.

Ejemplo Output :

Case 1: 6

Case 2: 1

Métodos

En el siguiente código podemos observar como se recoge los parámetros del input y se van almacenando en un vector para luego realizar la concatenación de los dos lenguajes guardando las cadenas resultantes en un set, para así eliminar los posibles resultados iguales y al finalizar imprimimos la longitud del set que es el número de cadenas resultantes.

```
1 def main():
2     T = int(input())
3     Case = []
4     while (T > 0):
5         result = set()
6         A = []
7         B = []
8         M, N = input().split()
9         M, N = [int(M), int(N)]
10        while (M > 0):
11            A.append(input())
12            M -= 1
13        while (N > 0):
14            B.append(input())
15            N -= 1
16        for x in A:
17            for y in B:
18                dummy = x + y
19                result.add(dummy)
20        Case.append(len(result))
21        T -= 1
22    for i in Case:
23        print("Case " + str(Case.index(i) + 1) + ": " + str(i))
24
25 if __name__ == '__main__':
26     main()
```

Resultados

La siguiente imagen muestra un test descrito en el problema comprobando que efectivamente, el output es el correcto.

```
th3ent:~/workspace/concurso-de-programacion-acm-icpc-problema-10887/src (master) $ python3 program.py
2
3 2
cat
dog
mouse
ratt
bat
1 1
abc
cab
Case 1: 6
Case 2: 1
```

La siguiente imagen es una captura de la web Uva Online Judge, con el resultado de las pruebas y el tiempo de ejecución promedio de las mismas(0.400).

| # | Problem | Verdict | Language | Run Time | Submission Date |
|----------|----------------------------------|----------|----------|----------|---------------------|
| 20343513 | 10887 Concatenation of Languages | Accepted | PYTH3 | 0.400 | 2017-11-14 09:58:29 |

Como se puede observar, el código fue aceptados(es decir, pasó todas las pruebas realizadas)

Uno de los requisitos de la web era que el tiempo de ejecución de tu código fuera inferior a 3.00 segundos

| Total Submissions | | Users that tried it | | Users that solved it |
|------------------------|------------|---------------------|----------|----------------------|
| 10096 | | 1483 | | 988 |
| Your best accepted try | | | | |
| Ranking | Submission | Run Time | Language | Submission Date |
| 86 | 20343513 | 0.400 | PYTH3 | 2017-11-14 09:58:29 |

En la imagen superior, se puede ver nuestra posición final en el ranking (86 de 988), así como el tiempo de ejecución promedio (0.400), el lenguaje seleccionado (PYTH3) y la fecha en la que se subió el programa al sistema (2017-11-14 09:58:29)

Conclusiones

Para la realización de este proyecto decidimos programar usando python3 debido a su facilidad a la hora de escribir el código y la sencillez del mismo. Tras comparar nuestros resultados con los del top del IACM-ICPC, podemos concluir que python, no es el lenguaje más adecuado cuando se quiere optimizar al máximo el tiempo de ejecución de un programa para participar en competiciones de este estilo en las que lo importante es la velocidad, y se compara solo eso y no se hace diferencia por categorías de los diferentes lenguajes. Aunque sencillo, nuestro código es interpretado, y este proceso no puede competir con lenguajes compilados mucho más rápidos a la hora de ejecutar como c o c++, como podemos ver en la siguiente imagen:

| Top 20 | | | | | |
|---------|------------|---------------------------------|----------|----------|---------------------|
| Ranking | Submission | User | Run Time | Language | Submission Date |
| 1 | 16655956 | M4G!C!4N | 0.003 | ANSI C | 2016-01-01 08:35:09 |
| 2 | 8035154 | cpu | 0.008 | ANSI C | 2010-06-14 10:57:49 |
| 3 | 6613387 | Arturo Aguirre Escobar | 0.010 | C++ | 2008-08-26 14:05:01 |
| 4 | 11288873 | kenethyph | 0.012 | ANSI C | 2013-02-14 15:28:55 |
| 5 | 19857925 | 黄汉升 | 0.020 | C++11 | 2017-08-15 05:57:26 |
| 6 | 3814677 | Ivan Krasilnikov | 0.025 | ANSI C | 2005-08-08 08:54:17 |
| 7 | 19961978 | Zubayet Zaman Zico | 0.030 | C++11 | 2017-09-05 10:01:11 |
| 8 | 3819445 | Constantin Jucovschi | 0.061 | C++ | 2005-08-09 14:34:00 |
| 9 | 3823118 | protik mohammad hossain | 0.061 | C++ | 2005-08-10 14:57:26 |
| 10 | 5161167 | Willebaldo G | 0.064 | ANSI C | 2006-11-26 05:50:01 |
| 11 | 17831058 | 赵子龙 | 0.070 | C++11 | 2016-08-13 04:36:56 |
| 12 | 17827758 | Jim | 0.080 | C++ | 2016-08-12 11:40:21 |
| 13 | 3826386 | tumkoepfe 2003 | 0.082 | C++ | 2005-08-11 14:40:40 |
| 14 | 15934838 | Bogdan Ciobanu | 0.089 | C++ | 2015-08-14 13:32:31 |
| 15 | 17378546 | Sam | 0.090 | C++11 | 2016-05-18 07:20:50 |
| 16 | 17961413 | bnuvjudge4 | 0.090 | C++ | 2016-09-06 12:43:46 |
| 17 | 16151171 | Tom | 0.093 | C++11 | 2015-09-23 13:21:27 |
| 18 | 16151328 | 关云长 | 0.106 | C++11 | 2015-09-23 13:47:13 |
| 19 | 3836541 | Marcin Wielgus | 0.110 | C++ | 2005-08-14 23:45:05 |
| 20 | 4090933 | Yi-kwon Hwang - Just for fun :) | 0.110 | C++ | 2005-11-01 05:42:21 |

Referencias

UVa Online Judge url: <https://uva.onlinejudge.org/>

IACM-ICPC url: <https://icpc.baylor.edu/>

Python3 url: <https://www.python.org/>

Github url: <https://github.com/>

Cloud 9 url (entorno de desarrollo usado): <https://c9.io>