



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283
Tarea 3
Entrega: 26 de noviembre (hasta las 11:59pm)

La solución de cada problema debe ser un programa en Python 3. Estos programas deben ser entregados a través del Siding. El formato de entrega será informado en el foro del curso en el Siding. En este foro además puede hacer preguntas sobre los problemas de la tarea. Si quiere usar alguna librería en sus soluciones debe preguntar en el foro del curso si esta librería está permitida. El foro es el canal de comunicación oficial para todas las tareas.

Problema 1

Dado un string $S = s_1 s_2 \cdots s_n$, su programa debe encontrar la secuencia de caracteres $P = c_1 \cdots c_k$ más larga que aparezca más de una vez en S . Si existe más de un P de largo máximo que cumpla con esto, entregue el lexicográficamente menor entre ellos.

Formalmente, decimos que un string $P = c_1 \cdots c_k$ calza desde ℓ en un string $S = s_1 s_2 \cdots s_n$ si $\ell + k - 1 \leq n$ y para todo $i \in [1, k]$ se tiene que $s_{\ell+i-1} = c_i$. Decimos que P aparece más de una vez en S si existen $\ell_1, \ell_2 \in [1, n - k + 1]$ tales que $\ell_1 \neq \ell_2$ y que P calza en S desde ℓ_1 y desde ℓ_2 .

Importante: El string S es generado aleatoriamente. Precisamente, en cada test se fija un n , y se prueba con un string S elegido con distribución uniforme sobre Σ^n , donde Σ son las letras minúsculas del alfabeto inglés de la **a** a la **z**.

Formato

El input consistirá de una línea con el string $S = s_1 s_2 \cdots s_n$.

Su output debe ser una sola línea con el string P .

Límites

$$27 \leq n \leq 10^5$$

s_i es una letra minúscula del alfabeto inglés de la **a** a la **z**, para cada $i \in [1, n]$

Tiempo de ejecución

2 segundos

Complejidad esperada

$O(k^2 \times n)$. Para este problema se espera un algoritmo **determinista** que sea eficiente sobre un input que es generado de manera aleatoria con distribución uniforme (en el espíritu de Quicksort).

Ejemplos

Input 1

```
xknovhovysdzgwlbswtasiicvo
```

Output 1

```
ii
```

Input 2

```
kodxjmultfmweuqpenmjnsjgkswsokodxvobgqvpzcqhdaosi
```

Output 2

```
kodx
```

Problema 2

Dados N puntos en \mathbb{R}^3 y un plano P en \mathbb{R}^3 , se dice que un radio $R \in \mathbb{R}$ ($R \geq 0$) es válido si es que existe un punto $X \in \mathbb{R}^3$ ($X \notin P$) tal que si se lanza un rayo con origen en X y dirección perpendicular al plano P , el rayo en su trayecto atraviesa a N esferas, donde la i -ésima esfera está centrada en el i -ésimo punto y su radio es R . En esta pregunta usted recibirá como entrada N puntos y M planos, y por cada plano se le pide encontrar el mínimo radio R válido asociado.

Formato

La primera línea contiene dos enteros positivos N y M , indicando la cantidad de puntos y la cantidad de planos respectivamente. Luego vienen N líneas cada una con tres enteros x_i, y_i, z_i correspondientes a las coordenadas del i -ésimo punto. Finalmente siguen M líneas cada una con tres enteros a_i, b_i, c_i correspondientes a los coeficientes de la ecuación del i -ésimo plano ($a_i x + b_i y + c_i z = 0$).

Su output debe ser M líneas. La i -ésima línea debe contener el valor del mínimo radio R válido asociado al i -ésimo plano, como se definió en el enunciado. Su respuesta se considerará correcta si los errores relativos y absolutos no exceden a 10^{-6} . Entiéndase por error absoluto $|X_{\text{real}} - X_{\text{pred}}|$ y por error relativo $|X_{\text{real}} - X_{\text{pred}}|/|X_{\text{real}}|$.

Límites

$$1 \leq N \leq 5000$$

$$1 \leq M \leq 400$$

$$-10^5 \leq x_i, y_i, z_i \leq 10^5, i \in [1, N]$$

$$-200 \leq a_i, b_i, c_i \leq 200, a_i^2 + b_i^2 + c_i^2 > 0, i \in [1, M]$$

Tiempo de ejecución

3 segundos

Complejidad esperada

$O(N \times M)$. Para este problema se espera un algoritmo **aleatorizado** cuya complejidad esperada sea la señalada.

Ejemplos

Input 1

```
3 1
0 1 0
0 2 0
0 3 0
0 1 0
```

Output 1

```
0.0000000000
```

Input 2

```
7 5
10000 10000 10000
2 0 0
2 2 0
0 2 1000
0 0 10
1 1 1
1000 1000 1000
1 0 0
0 1 0
0 0 1
1 1 1
1 1 0
```

Output 2

```
7071.0678118655
7070.3607404432
7071.0678118655
408.6571505145
5000.0000500000
```

Problema 3

Hay N personas reunidas con el objetivo de escoger un comité representativo. Para ello se presentan C candidatos, de los cuales se escogerá un subconjunto (posiblemente vacío) para conformar el comité.

Cada persona tiene preferencias: hay candidatos que le caen bien y candidatos que le caen mal (y el resto le es indiferente). Cada persona quiere que los candidatos que le caen bien estén en el comité, y que los que le caen mal no estén.

Dada una lista de candidatos y las preferencias de cada persona, se le pide encontrar un comité que satisfaga **estrictamente más de** $1/3$ de las preferencias de cada persona. Se entiende que una preferencia es satisfecha: (1) si candidato X me cae mal y no está en el comité, o bien (2) si candidato X me cae bien y sí está en el comité.

Importante: Puede asumir que siempre existe solución. Más aún, puede asumir que existe una solución que satisface **al menos** $2/3$ de las preferencias de cada persona.

Formato

La primera línea contiene un entero positivo N indicando la cantidad de personas. Luego vienen N líneas. La i -ésima línea consiste en un entero K_i seguido de K_i palabras separadas por espacio. Cada palabra comienza con un '+' (favor) o un '-' (rechazo), seguido del nombre de un candidato (hasta 20 letras en minúscula del abecedario inglés), indicando las preferencias de la i -ésima persona.

Su output debe ser una sola línea con una lista de nombres de candidatos, separados por espacio, que usted propone para conformar el comité. Su respuesta será considerada correcta si es que los candidatos mencionados están todos presentes en el input, si no se repiten candidatos y si por cada persona se satisface **estrictamente más de** $1/3$ de sus preferencias.

Límites

$$1 \leq N \leq 10000$$

$$1 \leq K_i \leq 30, i \in [1, N]$$

$C \leq 250$, donde C es la cantidad de candidatos distintos.

El nombre de un candidato no excede los 20 caracteres.

Tiempo de ejecución

3 segundos

Complejidad esperada

$O(N \times K + C^2)$. Para este problema se espera un algoritmo **aleatorizado** que converja a la respuesta correcta en alrededor de C^2 iteraciones.

Ejemplos

Input 1

1

4 +juan +josefa +maria -pedro

Output 1

juan josefa maria pedro

Este comité es válido ya que satisface $3/4$ (más de $1/3$) de las preferencias de la única persona.

Output 1 (alternativo)

juan josefa maria

Este comité es válido ya que satisface el 100% (más de $1/3$) de las preferencias de la única persona.

Input 2

3

3 +peter +john -mary

3 +peter -john +mary

3 -peter +john +mary

Output 2

peter john mary

Este comité es válido ya que satisface $2/3$ (más de $1/3$) de las preferencias de las 3 personas.

Problema 4

Dado un arreglo de enteros $A = [a_1, a_2, \dots, a_n]$, queremos encontrar un entero $d > 0$ que cumpla con las siguientes condiciones:

- Existe un $T \subseteq [1, n]$ tal que $|T| \geq n/3$ y para cada $i \in T$ se cumple que $d \mid a_i$.
- Para cualquier d' que cumple la condición anterior, se tiene que $d' \leq d$.

Su programa debe recibir este arreglo A y entregar d .

Otra forma de ver este problema, es que la primera condición define un d' para cada subconjunto $T' \subseteq [1, n]$ de tamaño mayor o igual a $n/3$. Su programa debe entregar el mayor de ellos.

Importante: De manera excepcional, para esta pregunta el código que entregue el alumno será corrido varias veces en cada caso de prueba, y el caso dará puntaje (y se dejará de revisar) si en al menos una de sus ejecuciones, el output es el correcto. No obstante, si antes de encontrar un resultado correcto la ejecución excede el tiempo límite, el caso se dejará de revisar y no dará puntaje. En otras palabras, para tener puntaje en un caso, tiene que haber una ejecución que entregue el resultado correcto y corra en el límite de los 2 segundos, y todas las ejecuciones anteriores tienen que haber tardado, cada una, menos de 2 segundos.

Formato

El input consistirá de dos líneas separadas por espacios que describen el arreglo A :

n

$a_1 \cdots a_n$

Su output debe ser una sola línea con el número d .

Límites

$$1 \leq n \leq 10^5$$

$$1 \leq a_i \leq n \text{ para cada } i \in [1, n]$$

Tiempo de ejecución

2 segundos

Complejidad esperada

$O(n^{1.5})$. Para este problema se espera un algoritmo **aleatorizado** con baja probabilidad de error.

Ejemplos

Input 1

6
2 5 6 3 2 3

Output 1

3

Input 2

12
1 5 10 4 8 9 12 7 5 5 12 8

Output 2

5

Hint

Un número n tiene $O(\sqrt{n})$ divisores. El número con más divisores en el rango $[1, 10^5]$ es 83160.