Problema D

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 313.25MB |
| Límite de tiempo (caso) | 3s | Límite de tiempo (total) | 60s |

**Descripción**

Limak es un pequeño oso polar. Hoy se encontró con algo delicioso en la nieve. Es una barra cuadrada de chocolate con NN XX NN pedazos. Algunas piezas son especiales porque hay cerezas en ellas.

Limak va a usar sus garras para romper el chocolate en dos partes. Se puede hacer un corte entre dos filas o entre dos columnas. Esto significa que no puede romper cualquier pieza.

Limak va a comer una parte en este momento y la segunda parte para mañana. Las cerezas son muy importantes para él, así que quiere tener el mismo número de cerezas hoy y mañana. Aunque las partes no necesariamente tienen que tener el mismo número de trozos de chocolate.

Teniendo en cuenta la descripción de un chocolate, ¿Se puede comprobar si Limak puede hacer un corte que divide un chocolate en dos partes con igual número de cerezas?

Nota:Nota: Está garantizado que un chocolate contiene al menos una cereza.

**Entrada**

La primera línea de la entrada es un número TT que representa la cantidad de casos de prueba. Cada caso de prueba inicia con una línea con un entero NN que representa el tamaño del lado de la barra de chocolate. Cada una de las siguientes NN líneas contiene una cadena de tamaño . Cada carácter de una cadena es o ‘#’ (cereza) o ‘.’(Pieza vacía). En cada caso de prueba al menos un carácter es ‘#’.

**Salida**

Por cada caso se debe imprimir la palabra ‘YES’ si Limak puede romper el chocolate de acuerdo a las reglas descritas antes. De lo contrario, imprimir ‘NO’

NOTA:NOTA: Tenga en cuenta la dimensión máxima del problema, evite realizar cálculos repetitivos en su solución

**Ejemplo**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| 2  3  ###  ##.  ###  4  ####  .##.  .#..  #... | NO  YES | casocaso 1:1: Es imposible hacer un corte que permita tener 4 cerezas a cada lado.  casocaso 2:2: Se puede cortar entre la primera y la segunda fila para conseguir 4 cerezas en cada lado del chocolate. |

**Límites**

* 0<T<10000<T<1000
* 1≤N≤10001≤N≤1000

*Fuente: Estructuras de Datos Universidad Nacional de Colombia*

Problema subido por: [Christian Zuluaga](https://omegaup.com/profile/cmzuluagam/)

| **Enviado** | **GUID** | **Status** | **Porcentaje** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Problema-D#problems/new-run) | | | | | | | | |
| 2016-11-06 22:32:47 | d1316535 | Respuesta correcta | 100.00% | cpp | 4.03 MB | 0.12 s |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Problema-D#problems>

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

int main() {

  int t;

  scanf("%d", &t);

  while(t--) {

      //------ENTRADA--------------------------------

         int n;

         scanf("%d", &n);

         string filas[n];

         for(int i =0; i < n; i++) {

            string f;

            cin >> f;

            filas[i] = f;

         }

       //--------ALGORITMO------------------------------

        std::vector<int> sumfilas;

        std::vector<int> sumcols;

*/\**

*para no repetir calculos y pasar varias veces por la matriz,*

*como los cortes son por filas y por columnas, es decir, se hace un solo corte sobre una fila,*

*o se hace un solo corte sobre una columna, entonces hago toda la sumatoria de veces que aparece #*

*de cada fila y la guardo en el vector sumfilas,*

*a su vez hago toda la sumatoria de las veces que aparece # en cada columna, y las guardo*

*en sumcols\*/*

        int totalcerezas =0;

        for(int i =0; i < n; i++) {

           int sf =0, sc =0;

           for(int j =0; j < n; j++) {

              if(filas[i][j] == '#') {

                sf++;

                totalcerezas++;

              }

              if(filas[j][i] == '#') {

                sc++;

              }

           }

           sumfilas.push\_back(sf);

           sumcols.push\_back(sc);

        }

        std::string ans = "NO";

*/\**

*como ya tengo el total de cerezas, los voy restando por filas y por columnas,*

*en sum acumulo lo que hay hasta ahi de filas o de columnas, y le resto a totalcerezas*

*para saber cual es el resto, entonces si el resto es igual a sum es porque hay 2 lados*

*con la misma cantidad de cerezas*

*\*/*

        int sumporfila=0, sumporcol=0;

        for(int i =0; i<n; i++) {

           sumporfila += sumfilas[i];

           sumporcol += sumcols[i];

           if(totalcerezas - sumporfila == sumporfila || totalcerezas - sumporcol == sumporcol) {

               ans = "YES";

               break;

            }

        }

        cout << ans << endl;

  }

  //  system("pause");

 return 0;

}

---- Y SOY EL QUE MENOS TIEMPO Y MEMORIA USÓ PARA ESTE PROBLEMA--------

| **Usuario** | **Lenguaje** | **Memoria** | **Tiempo** | **Enviado** |
| --- | --- | --- | --- | --- |
| [nacho\_M](https://omegaup.com/profile/nacho_M) | cpp | 3.95 | 0.12 | 2016-11-06 22:59:13 |
| [davides123](https://omegaup.com/profile/davides123) | py | 7.51 | 0.39 | 2016-09-04 20:38:49 |
| [MiguelMedellin](https://omegaup.com/profile/MiguelMedellin) | java | 25.93 | 1.59 | 2016-08-30 21:37:58 |
| [FelipeVillamil](https://omegaup.com/profile/FelipeVillamil) | java | 21.98 | 1.60 | 2016-08-20 00:40:46 |
| [djsilverav](https://omegaup.com/profile/djsilverav) | java | 25.21 | 1.62 | 2016-08-19 16:32:07 |
| [cglaitong](https://omegaup.com/profile/cglaitong) | java | 24.24 | 1.62 | 2016-08-23 12:12:36 |
| [acmartinn](https://omegaup.com/profile/acmartinn) | java | 25.91 | 1.69 | 2016-08-22 02:05:45 |
| [dcastanedar](https://omegaup.com/profile/dcastanedar) | java | 23.11 | 1.70 | 2016-08-23 18:49:26 |
| [DiegoChavarro](https://omegaup.com/profile/DiegoChavarro) | java | 21.98 | 1.71 | 2016-08-24 00:40:44 |
| [cdcaballeroa](https://omegaup.com/profile/cdcaballeroa) | java | 25.07 | 1.73 | 2016-08-21 22:22:40 |
| **Mejores envíos aceptados** | | | | |