

Fundamentos de Algoritmos. Grupo E

Práctica 3: Vuelta atrás

12 de Diciembre de 2024

1. Debéis entregar la práctica durante la hora de laboratorio.
2. En el campus disponéis de un fichero plantilla para hacer la práctica. **Usadla, con eso evitaréis errores innecesarios.**
3. **IMPORTANTE:** asegúrate de que el Visual Studio tiene configurado el **fin de línea como CRLF**
4. Escribid el **nombre y apellidos** de los dos miembros de la pareja que estéis haciendo la práctica. Si no los incluís, el ejercicio se evaluará con un 0.
5. Debéis desarrollar e implementar una solución eficiente y entregarla en el juez automático. **Sólo es necesario que suba el código uno de los dos miembros.**
6. Vuestra solución será evaluada por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayáis realizado como pareja.
7. Durante el tiempo de laboratorio, en el juez únicamente está subido el caso de prueba del enunciado. Esto significa que un resultado de **correct** es **provisional**.
8. A lo largo del examen se os pedirá que os identifiquéis y rellenéis vuestros datos en una hoja de firma.

Un convite por compromiso

Dicen que una boda es un motivo de alegría pero a Felisa Amparo Limón se está convirtiendo en un suplicio. El problema no es la propia boda sino el dichoso convite y los compromisos familiares.

El primer problema es que la familia de Felisa es demasiado numerosa y peculiar, mientras que los huecos para el convite es más reducido, por lo que no puede invitar a todo el mundo. Afortunadamente, Felisa conoce muy bien a sus familiares, por lo que sabe estimar la satisfacción que de cada familiar tendrá por sentarse en uno de los tan codiciados puestos de invitados.

El segundo problema es la tía Remedios. Es una pesada y se ha empeñado en pagar el convite entero como estrategia para asegurarse de que la invitan sí o sí.

La meta de Felisa es maximizar la satisfacción de los asistentes al convite para que el festejo sea lo más feliz posible cumpliendo las siguientes condiciones:

- A ningún invitado (incluyendo a la tía Remedios) se le asignará un puesto para el que tenga satisfacción negativa. En cualquier caso preferirá no ir antes que ser invitado para sentarse en un lugar que no le satisfaga.
- La familia de Felisa es muy supersticiosa y está convencida de que es de mal augurio sentarse en el puesto cuyo número coincide con el número de la lista general. Por lo tanto, Felisa sólo quiere soluciones que, a lo sumo, asigne un tercio de los puestos totales a invitados cuyo número en la lista sea igual al del puesto. Al calcular el tercio de los puestos se truncará el resultado.
- La tía Remedios debe ser incluida entre los invitados. No es necesario que ocupe ningún puesto especial.

Felisa no sólo está interesada en la máxima satisfacción posible sino también en el número de soluciones que dan esa satisfacción máxima.

Escribe un algoritmo con la técnica de vuelta atrás. Describe también el espacio de soluciones y los marcadores usados. Indica e implementa una función de poda.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de varias líneas. En la primera se muestra el número de familiares que pueden ser invitados al convite ($N > 0$), el número de invitados que vamos a seleccionar (coincide con el número de huecos) ($M > 0$) y el puesto de la lista de familiares en el que está la tía Remedios ($0 \leq T < N$).

Las siguientes M líneas se da la matriz de satisfacción (S) de cada persona por puesto del convite. Es decir, $S[i][j]$ indica la satisfacción de la persona i por sentarse en el puesto j del convite. A mayor sea el valor, más satisfacción. Si la satisfacción es negativa esa persona no querrá sentarse en ese puesto.

El final de datos se marca con un cero.

Salida

Para cada caso de prueba se escribe en una línea la satisfacción mejor que se puede conseguir y el número de combinaciones posibles de invitados que tenemos con esa satisfacción mayor. Si no existe solución se escribirá No.

Entrada de ejemplo

```
4 3 2
8 -1 3 5
-2 10 5 3
3 3 3 5
5 2 0
-1 5 5 5 5
-4 3 3 3 3
3 3 2
10 -1 2
2 10 -6
-3 2 10
3 3 2
10 2 2
2 10 2
2 2 10
0
```

Salida de ejemplo

```
18 2
No
6 1
14 3
```