OIE 2026

XXX Olimpiada Informática Española

Soluciones a los problemas

Comité científico OIE

Valladolid, 10–12 abril 2026

Destripando los problemas

- ¡Esperamos que os hayan gustado los problemas!

Destripando los problemas

- ¡Esperamos que os hayan gustado los problemas!
- **Un reto exigente:** Sabemos que han sido horas de mucha tensión y desgaste mental. ¡Enhorabuena a todos por el esfuerzo!

Destripando los problemas

- ¡Esperamos que os hayan gustado los problemas!
- **Un reto exigente:** Sabemos que han sido horas de mucha tensión y desgaste mental. ¡Enhorabuena a todos por el esfuerzo!
- **Variedad algorítmica:** Hemos intentado equilibrar la balanza tocando grafos, combinatoria y algoritmos voraces, ¡sin olvidar formatos especiales como el interactivo y el *output-only*!

Destripando los problemas

- ¡Esperamos que os hayan gustado los problemas!
- **Un reto exigente:** Sabemos que han sido horas de mucha tensión y desgaste mental. ¡Enhorabuena a todos por el esfuerzo!
- **Variedad algorítmica:** Hemos intentado equilibrar la balanza tocando grafos, combinatoria y algoritmos voraces, ¡sin olvidar formatos especiales como el interactivo y el *output-only*!
- **Soluciones completas:** Hoy veremos la "idea feliz" de cada problema. Los editoriales y códigos se publicarán próximamente en la web de la OIE.

Puntos y rayas (MORSE)

Enunciado

Dada una secuencia de dígitos, queremos saber cuántos puntos y cuántas rayas hay en total en su codificación en código morse.

Restricciones

- $1 \leq T \leq 1000$ casos de prueba.
- La longitud de la secuencia varía entre 1 y 1000.

Puntos y rayas (MORSE)

Ejemplo de Entrada y Salida

3 5 10 314159	5 0 1 9 15 15

Puntos y rayas (MORSE)

- Cada dígito decimal se representa con **exactamente 5 símbolos**.

Dígito	Código morse		Dígito	Código morse
0	-----		5	.....
1	.----		6	-....
2	..---		7	--...
3	...--		8	---..
4	....-		9	----.

Puntos y rayas (MORSE)

- Cada dígito decimal se representa con **exactamente 5 símbolos**.

Dígito	Código morse		Dígito	Código morse
0	`-----`		5	`.....`
1	`.----`		6	`-....`
2	`..---`		7	`--...`
3	`...--`		8	`---..`
4	`....-`		9	`----.`

- Podemos observar que el número de puntos para un dígito d es $\min(d, 10 - d)$.

Puntos y rayas (MORSE)

- Cada dígito decimal se representa con **exactamente 5 símbolos**.

Dígito	Código morse		Dígito	Código morse
0	-----		5	.....
1	.----		6	-....
2	..---		7	--...
3	...--		8	---..
4	....-		9	----.

- Podemos observar que el número de puntos para un dígito d es $\min(d, 10 - d)$.
- Como sabemos la longitud total en símbolos de la secuencia ($5 \times$ longitud), las rayas se sacan por diferencia.

Puntos y rayas (MORSE)

Solución en $O(N)$

$$\text{puntos} = \sum_{d \in s} \min(d, 10 - d)$$

$$\text{rayas} = 5 \times \text{longitud}(s) - \text{puntos}$$

Pagos al final del viaje (PAGOS)

Enunciado

Un grupo de n voluntarios hace m compras. Se busca un algoritmo que calcule quién debe pagar a quién para saldar todas las deudas, usando **como máximo** $n - 1$ **transferencias**.

Restricciones

- Hasta $n \leq 50.000$ voluntarios y $m \leq 50.000$ compras.
- El gasto total es siempre múltiplo de n (se puede dividir en partes iguales exactas sin decimales).

Pagos al final del viaje (PAGOS)

Ejemplo de Entrada y Salida

4 1 Jacobo Marco Pedro Joan Jacobo 400	3 Joan Jacobo 100 Pedro Jacobo 100 Marco Jacobo 100

Pagos al final del viaje (PAGOS)

Ejemplo de Entrada y Salida

2 3 Jacobo Marco Jacobo 1 Jacobo 2 Marco 3	0

Pagos al final del viaje (PAGOS)

Ejemplo de Entrada y Salida

3 3 dario manu alberto dario 4 manu 5 alberto 6	1 dario alberto 1

Pagos al final del viaje (PAGOS)

- Primero calculamos la cuota ideal por persona: gasto total / *n*.

Pagos al final del viaje (PAGOS)

- Primero calculamos la cuota ideal por persona: gasto total / *n*.
- Calculamos el balance de cada persona y los dividimos en dos grupos: **deudores** (pagaron de menos) y **acreedores** (pagaron de más).

Pagos al final del viaje (PAGOS)

- Primero calculamos la cuota ideal por persona: gasto total / *n*.
- Calculamos el balance de cada persona y los dividimos en dos grupos: **deudores** (pagaron de menos) y **acreedores** (pagaron de más).
- **Algoritmo Voraz (Greedy):** Emparejamos un deudor con un acreedor y transferimos el máximo posible entre ellos.

Pagos al final del viaje (PAGOS)

¿Por qué garantiza como mucho $n - 1$ transferencias?

$$\text{Transferencia} = \text{mín}(\text{deuda}, \text{crédito})$$

En cada paso, al menos una persona salda su cuenta por completo y sale de la lista. Como hay n personas y la suma total de saldos es 0, en el peor de los casos haremos $n - 1$ pasos para dejar a todo el mundo a cero.

La Picasso de Mnieto (COCHE)

Enunciado

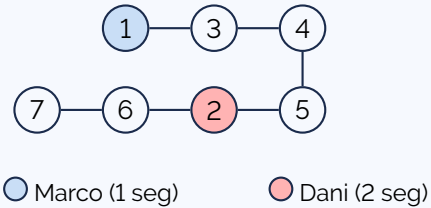
Dani (nodo 2) camina a 2 seg/arista. Su hermano Marco (nodo 1) conduce a 1 seg/arista. Dani puede encontrarse con Marco en cualquier nodo para subirse al coche. ¿Cuál es el tiempo mínimo para que Dani llegue a cada nodo del grafo?

Restricciones

- Grafo no dirigido conexo de $n \leq 10^5$ nodos y $m \leq 2 \cdot 10^5$ aristas.
- La suma de n no supera 10^5 y la de m no supera $2 \cdot 10^5$.

La Picasso de Mnieto (COCHE)

Primer caso de prueba

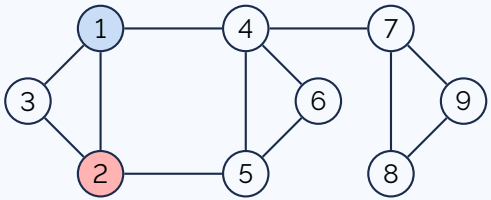


Salida:

6 0 5 4 2 2 4

La Picasso de Mnieto (COCHE)

Segundo caso de prueba (subtarea 1)

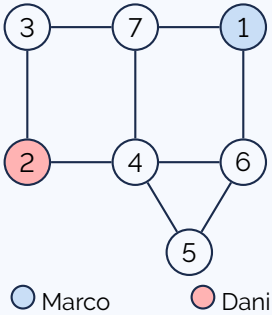


Salida:

2 0 2 3 2 3 4 5 5

La Picasso de Mnieto (COCHE)

Tercer caso de prueba



Salida:

4 0 2 2 3 3 3

La Picasso de Mnieto (COCHE)

- Dani tiene dos opciones para llegar a un nodo: ir caminando directamente, o encontrarse con Marco en un nodo intermedio y llegar en coche.

La Picasso de Mnieto (COCHE)

- Dani tiene dos opciones para llegar a un nodo: ir caminando directamente, o encontrarse con Marco en un nodo intermedio y llegar en coche.
- Hacemos dos **BFS** (Búsqueda en Anchura) iniciales:
  - Desde el nodo 1 (Marco) para obtener D_M.
  - Desde el nodo 2 (Dani) para obtener D_D.

La Picasso de Mnieto (COCHE)

- Dani tiene dos opciones para llegar a un nodo: ir caminando directamente, o encontrarse con Marco en un nodo intermedio y llegar en coche.
- Hacemos dos **BFS** (Búsqueda en Anchura) iniciales:
  - Desde el nodo 1 (Marco) para obtener D_M.
  - Desde el nodo 2 (Dani) para obtener D_D.
- Si deciden encontrarse en un nodo i, el tiempo que tardan en reunirse es $\max(D_M[i], 2 \times D_D[i])$, ya que ambos deben haber llegado a ese nodo.

La Picasso de Mnieto (COCHE)

- Una vez que Dani está en el coche en un nodo intermedio, puede viajar a cualquier destino *v* a velocidad de coche (1 seg/arista).

La Picasso de Mnieto (COCHE)

- Una vez que Dani está en el coche en un nodo intermedio, puede viajar a cualquier destino v a velocidad de coche (1 seg/arista).

Algoritmo final en $O(E \log V)$

1. Inicializamos un **Dijkstra** metiendo en la cola todos los nodos i con su tiempo de encuentro inicial: $\max(D_M[i], 2 \times D_D[i])$.
2. Propagamos sumando 1 segundo por arista (velocidad de coche).
3. Para cada destino j, la respuesta óptima es el **mínimo** entre el resultado del Dijkstra en j y el tiempo de ir caminando ($2 \times D_D[j]$).

Camino a la OIE (AUTOBUSES)

Enunciado

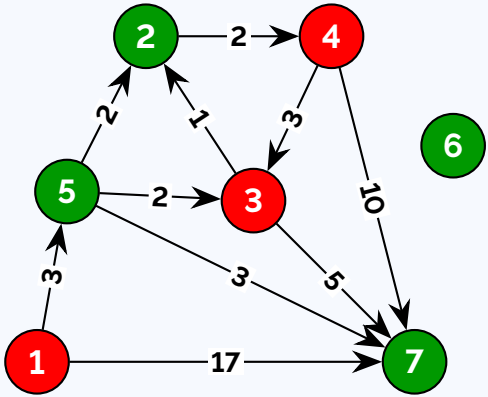
Ariadna y Darío viajan por una red de metro de n paradas y m líneas para llegar a la estación (parada n). En las paradas rojas decide Ariadna (que busca **minimizar** el tiempo) y en las verdes Darío (que busca **maximizarlo**). ¿Cuánto tardarán si ambos juegan de forma óptima?

Restricciones

- $n, m \leq 2 \cdot 10^5$.
- La suma de n y la suma de m sobre todos los casos es $\leq 2 \cdot 10^5$.
- Darío prefiere dar vueltas infinitas (tiempo INFINITO) a llegar.

Camino a la OIE (AUTOBUSES)

Ejemplo



Salida:

15

Camino a la OIE (AUTOBUSES)

- Vamos a calcular, desde cada nodo, cuanto tardarán si ambos juegan óptimamente.
- Desde el nodo $n - 1$ ya lo sabemos, la respuesta es 0
- Si un nodo es **verde**, la respuesta de ese nodo es el **máximo** de, para cada edge que sale, su peso más la respuesta del nodo al que lleva.
- Si un nodo es **rojo**, la respuesta de ese nodo es el **máximo** de, para cada edge que sale, su peso más la respuesta del nodo al que lleva.
- Podemos intentar hacer una especie de Dijkstra, teniendo en cuenta estos dos hechos.

Camino a la OIE (AUTOBUSES)

Solución

- Creas una priotity queue y una array d, en que guardaremos lo que se tarda en llegar al final desde ese nodo. Inicialmente metemos a la cola el nodo $n-1$ con distancia 0.
- Cuando procesamos un nodo V de la cola, si ya lo hemos visto lo ignoramos. Si no, guardamos su distancia y metemos otros nodos a la cola. Si hay un edge de U a V con peso w:
  - Si U es rojo, lo metemos con distancia $d[V] + w$.
  - Si U es verde, lo metemos solo si hemos procesado ya todos los edges que salen de U. En ese caso, lo metemos con distancia el máximo de $w_i + d[V_i]$ sobre los edges (V_i, w_i) que salen de U.

Baba has exam (EXAMEN)

Enunciado

Baba tiene nivel inicial w_i en n temas y necesita acertar k preguntas. Puede estudiar h_i horas (incluso decimales) para que su nivel pase a ser $w_i + h_i$. El profesor pondrá dificultades $d_i \geq 0$ tales que $\sum d_i = D$. Baba acierta si $w_i + h_i \geq d_i$. ¿Mínimo de horas a estudiar para **garantizar** k aciertos frente a **cualquier** examen?

Restricciones

- $n \leq 2 \cdot 10^5$, $\sum n \leq 2 \cdot 10^5$.
- Dificultad total $D \leq 1000$.
- La respuesta puede ser decimal (se requiere precisión de 10^{-4}).

Baba has exam (EXAMEN)

- La mejor estrategia para el profesor es un greedy. El profesor irá por los temas en que Baba tiene un nivel positivo de pequeño a grande y les podrás dificultad un poco más grande que el nivel de Baba.
- De esto deduces que, si $w_1 \geq \ldots \geq w_n$ son los niveles de Baba en las subtareas, Baba aprobará siempre el examen si y solo si

$$\text{máx}(w_k, 0) + \text{máx}(w_{k+1}, 0) + \ldots + \text{máx}(w_n, 0) \geq D$$

- Si fijamos $k \leq m \leq n$ como el máximo tal que $w_m \geq 0$, esto se reduce a

$$w_k + w_{k+1} + \ldots + w_m \geq D$$

- Si distribuimos las horas de estudio adecuadamente, podemos hacerlo sin que el orden relativo de los niveles de las asignaturas cambie tras estudiar. Además, podemos estudiar solo un rango contiguo de asignaturas..

Baba has exam (EXAMEN)

- La mejor estrategia para el profesor es un greedy. El profesor irá por los temas en que Baba tiene un nivel positivo de pequeño a grande y les podrás dificultad un poco más grande que el nivel de Baba.
- De esto deduces que, si $w_1 \geq \ldots \geq w_n$ son los niveles de Baba en las subtareas, Baba aprobará siempre el examen si y solo si

$$\max(w_k, 0) + \max(w_{k+1}, 0) + \ldots + \max(w_n, 0) \geq D$$

- Si fijamos $k \leq m \leq n$ como el máximo tal que $w_m \geq 0$, esto se reduce a

$$w_k + w_{k+1} + \ldots + w_m \geq D$$

- Si distribuimos las horas de estudio adecuadamente, podemos hacerlo sin que el orden relativo de los niveles de las asignaturas cambie tras estudiar. Además, podemos estudiar solo un rango contiguo de asignaturas..

Baba has exam (EXAMEN)

Solución

- Iteramos por todos los posibles valores de m, el número de asignaturas con nivel positivo tras entrenar, y cogeremos el mínimo de horas de estudio sobre todos los valores.
- Mantenemos un segundo puntero para r, la asignatura de nivel más alto que tenemos que estudiar para llegar a aprobar el examen fijado m. Con esto sabemos el rango que actualizamos.
- Si sabemos cuál es el rango que actualizamos, es fácil calcular cuanto tenemos que estudiar en $O(1)$ calculando la suma del rango con prefix sums. Si la suma es s, debemos estudiar $D - s$ horas.

Ladrones en el museo (LADRONES)

Enunciado

Un museo es un árbol de n salas. Los guardias empiezan en la sala 1 y deben explorar para arrestar a todos los ladrones, que se mueven libremente por las salas no vigiladas. Para evitar que escapen, **toda sala revisada conectada a una no revisada debe tener un guardia**. ¿Mínimo de guardias necesarios?

Restricciones

- $n \leq 10^5$, $\sum n \leq 10^5$.
- El grafo es siempre un árbol conexo.

Ladrones en el museo (LADRONES)

Solucion

- Si empezamos a explorar el subárbol de un hijo, siempre es óptimo terminar de exploarlo antes de pasar a otro hijo.
- Calculamos DP_{nodo} = Coste explorar subárbol de $nodo$. El coste de explorar un hijo es $DP_{hijo} + 1$ excepto para el último hijo.
- Deducimos que $DP_{nodo} = \text{máx}_{u \in \text{ hijos } nodo} DP_u$, excepto si el máximo aparece más de una vez que obtendremos $DP_{nodo} = \text{máx}_{u \in \text{ hijos } nodo} DP_u + 1$.

Canal con interferencias (OPERACION)

Enunciado

Una secuencia circular de n bits cambia cada milisegundo por interferencias. El nuevo valor de cada bit se calcula haciendo un XOR de sus dos vecinos inmediatos: $s'_i = s_{i-1} \oplus s_{i+1}$. ¿Cuál será la secuencia final tras k milisegundos?

Restricciones

- $n \leq 10^5$, $\sum n \leq 10^5$.
- ¡El tiempo k puede ser enorme! $k \leq 10^{18}$.
- La secuencia es cíclica ($s_0 = s_n$, $s_{n+1} = s_1$).

Canal con interferencias (OPERACION)

Solucion

- Podemos demostrar por inducción que si llamamos f a la transformación, $f^{2^r}(s)_i = s_{i-2^r} \bigoplus s_{i+2^r}$.
- Descomponemos k es base binaria y aplicamos esta función $\log_2 k$ veces en las posiciones donde k tiene un 1 en base binaria.

Festival de música (MUSICA)

Enunciado

Se van a reproducir n canciones, cada una con una nota a_i del público. Se produce un **subidón** cuando una canción tiene nota **estrictamente mayor** que la inmediatamente anterior. ¿Cuántas permutaciones posibles de las canciones generan **exactamente** k subidones?

Restricciones

- $n \leq 2000$, $\sum n \leq 2000$.
- Notas de las canciones $a_i \leq 10^9$.
- La respuesta puede ser muy grande, devolver módulo $10^9 + 7$.

Festival de música (MUSICA)

Solucion

- Ordenamos a de manera creciente.
- Calculamos $DP_{i,j}$ = Número de ordenaciones con exactamente j subidones usando los i primeros valores de a.
- Cuando insertamos a_{i+1} a los i primeros valores, esta valor sabemos que es mas grande que todos los ya considerados, asi que obtenemos las transiciones:
  $DP_{i,j} = (j > 0 ? DP_{i-1,j-1} \cdot (i - j - cnt) : 0) + DP_{i-1,j} \cdot (j + 1 + cnt)$
  Donde cnt es el número de veces que aparece el valor a_{i+1} en los i primeros valores.

Canal con interferencias (TABLERO)

Solucion

La solución de ≈ 45 puntos, colocando 1215 piezas, es ir iterando y si la paridad está mal, invertir el valor de la casilla a la derecha.

Solucion

Si se asigna 0 o 1 de forma aleatoria a los elementos de la primera columna se consiguen ≈ 90 puntos con ≈ 2750 piezas colocadas.

Solucion

Para llegar a 3147, actualizamos solo con las 10000 mejores configuraciones, teniendo en cuenta lo que generarían en las 15 siguientes filas.

Canal con interferencias (TABLERO)

Hay muchas más soluciones que consiguen puntuaciones altas o incluso la completa. Se publicará un tutorial más detallado mañana por la tarde en Discord.

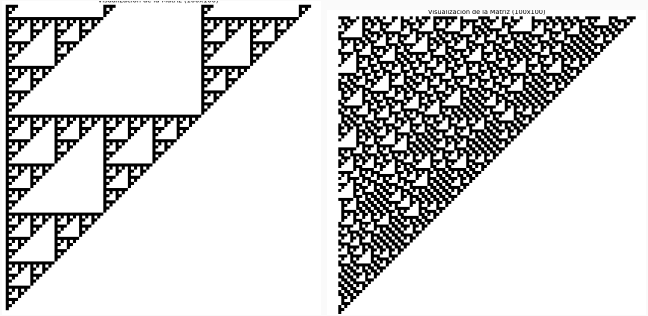


Figura 1: Visualización de los tableros con 1215 y 2762 1s.

Copia de seguridad corrupta (UNICOS)

Nota

La solución será publicada mañana por la tarde por el Discord de la OIE. ¡Uniros!