

Forza Mediana!

Ti viene fornito un set di n biglie, numerate da 0 ad $n - 1$, dove n è un numero dispari (ossia $n = 2k + 1$ per un qualche numero naturale k). Ciascuna biglia ha un peso, diverso dal peso di ogni altra biglia. Il tuo scopo è individuare la biglia mediana tra le $2k + 1$ biglie, ossia quella biglia che vede k biglie più leggere e k biglie più pesanti di sé stessa. Tu non hai accesso diretto alle biglie, ma puoi rivolgerti alla Grande Bilancia dei 3 Piatti, un rinomato oracolo che risponde a domande del tipo:

tra queste 3 biglie, quale è la biglia mediana.

Riesci a trovare la risposta che cerchi? Ti è richiesto di limitare, almeno asintoticamente, il numero di domande che poni all'oracolo.

Interazione

Il tuo programma interagirà col server (che tra l'altro fungerà anche da oracolo) leggendo dal proprio canale `stdin` e scrivendo sul proprio canale `stdout`. La prima riga di `stdin` contiene T , il numero di istanze (sets di palline) su cui evadere correttamente la domanda. Le istanze vanno quindi affrontate sequenzialmente, una ad una. Per ogni istanza, nella prima riga che leggi da `stdin` trovi il numero n di palline su cui lavorare. Se $n \geq 3$, si apre ora una fase in cui potrai porre le tue domande all'oracolo: per conoscere quale sia la mediana tra tre palline di nomi a, b, c in $[0, n - 1]$, scrivi su `stdout` una riga che riporti questi tre numeri separati da spazio; quindi leggi da `stdout` il nome della pallina mediana tra queste 3. Quando ti senti sicuro di conoscere il vero nome m della pallina mediana tra le n palline assegnate, scrivi su una riga di `stdout` il numero m ; e poi passa direttamente ad affrontare la prossima istanza.

Nota: Non sarà solo una questione di correttezza del tuo programma e di rispetto del time limit, ma anche di contenere il numero delle questioni che poni all'oracolo. Il criterio cui ti converrà guardare è quello del caso peggiore. Ma non cercare di minimizzare tale numero in modo esatto, la sua dipendenza funzionale da n è quasi sicuramente troppo complessa per essere conoscibile. Puoi invece porti l'obiettivo di contenere la crescita asintotica in n del numero di domande che porrai all'oracolo nel caso peggiore.

Esempi di interazioni possibili

Le righe che iniziano con `>` sono quelle inviate dal server, quelle che iniziano con `<` sono quelle inviate dal client. (Ignora però i commenti, ossia quella parte della riga che comincia col primo carattere di cancelletto `#` in essa eventualmente contenuto).

```
> 2      # numero di istanze/sets di biglie
> 3      # il numero n di biglie in questione per la prima istanza
< 1 0 2 # il problem solver (o il suo programma) pone una domanda all'oracolo
> 2      # l'oracolo risponde che la biglia labellata 2 è la mediana tra le biglie 0,
1, 2
< 2      # il problem solver consegna la sua risposta (corretta) per la prima istanza
> 5      # il numero n di biglie per la seconda istanza
< 0 1 2 # prima query all'oracolo
> 0      # 0 è la mediana tra 0, 1, 2
< 0 3 4 # 0 è la mediana tra 0, 3, 4, in realtà ora il problem solver fortunello conosce
già la risposta!
< 1 3 4 # ma ciò non gli impedisce di fare ulteriori domande
> 3      # 3 è la mediana tra 1, 3, 4
< 0      # ok, anche qui la risposta è corretta e il numero delle domande è comunque
rimasto basso
```

Subtask

Il tuo programma affronterà 50 istanze, che verranno generate per valori di n via via crescenti ($n = 99$ per la più grande). Il tempo limite per istanza (ossia per ciascun testcase) è sempre di 1 secondo.

Ma la vera sfida stà nel contenere la crescita asintotica (in n) del numero di domande, nel caso peggiore. Pertanto, il limite cui dovrai guardare maggiormente è quello di rimanere entro le 500 query (sulla singola istanza).

Se vuoi che il tuo programma si misuri su tutte le dimensioni di istanza previste, chiama:

```
rtal -s wss://ta.di.univr.it/algo connect forza_mediana -- python my_solution.py
```

Altrimenti, puoi evitare che il tuo programma venga condotto oltre un certo valore di n utilizzando l'argomento size. Ad esempio, per lanciare il programma sulle 4 istanze con numero di palline $n = 1, 3, 5, 7$:

```
rtal -s wss://ta.di.univr.it/algo connect forza_mediana -asize=7 -- python  
my_solution.py
```