به نام خدا

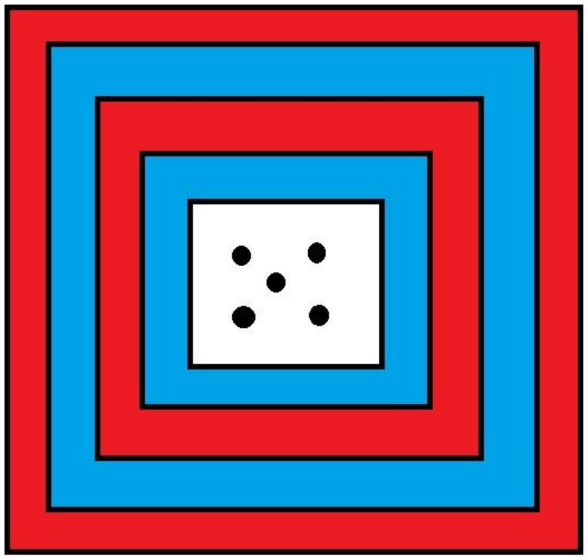
پاسخ پرسش‌های هفتگی نخست

22 دی 1391 خورشیدی

گروه چلنجر

راه حل پرسش نخست "مزرعه سلطان " :

می‌خواهیم ثابت کنیم پاسخ برابر $n(n+1)$ است. برای این منظور رنگ آمیزی زیر را در نظر می‌گیریم:



هر خانه در جدول با حداکثر دو خانه قرمز مجاور است. پس حداقل باید به اندازه نصف تعداد خانه‌های قرمز چاه داشته باشیم. تعداد خانه های قرمز برابر $2n(n+1)$ است(؟). پس حداقل به $n(n+1)$ چاه نیاز است. ارائه ساختار برای این عدد به عهده‌ی خودتان! ☺

راه حل پرسش دوم " ازدواج در خفنستان! " :

یک درخت دودویی در نظر می‌گیریم که در آن هر راس یک متغیر به عنوان شمارنده و 2 فرزند دارد. درخت در ابتدا فقط یک راس ریشه دارد (λ).

ما روی درخت دو عملیات مختلف انجام می‌دهیم :

**الف . اضافه کردن عدد x به درخت :**

ابتدا x را به مبنای 2 می‌بریم، a را برابر با پرارزش ترین رقم x قرار می‌دهیم و node را برابر با آدرس ریشه‌ی درخت.

در هر مرحله اگر a برابر با 0 بود node را برابر با آدرس فرزند سمت راست خودش قرار می‌دهیم ، در غیر این صورت برابر با آدرس فرزند سمت چپ خودش قرار می‌دهیم (در هرجای الگوریتم اگر می‌خواستیم به فرزند راسی برویم در حالی که هنوز فرزندی ندارد، آن‌ها را ایجاد می‌کنیم)، و در آخر شمارنده‌ی node را یکی زیاد می‌کنیم و a را برابر با رقم بعدی x قرار می‌دهیم.

**ب . پیدا کردن تعداد زوج‌های مناسب برای x :**

ابتدا k و x را به مبنای 2 می‌بریم، a را برابر با پر ارزش ترین رقم x و b را برابر با پر ارزش ترین رقم k قرار می‌دهیم و node را برابر با آدرس ریشه‌ی درخت. در هر مرحله

1 . اگر b برابر با صفر بود

1.1. اگر a صفر بود عدد شمارنده‌ی فرزند سمت چپ node را به جواب نهایی اضافه می‌کنیم و node را برابر با آدرس فرزند سمت راستش قرار می‌دهیم.

2.1. اگر a برابر با 1 بود، عدد شمارنده‌ی فرزند سمت node را به جواب نهایی اضافه می‌کنیم و node را برابر با آدرس فرزند سمت چپش قرار می‌دهیم.

2 . اگر b برابر با 1 بود

1.2. اگر a صفر بود، node را برابر با آدرس فرزند سمت چپش قرار می‌دهیم.

2.2. اگر a برابر با 1 بود، node را برابر با آدرس فرزند سمت راستش قرار می‌دهیم.

در انتها a وb را ترتیب برابر با رقم بعدی x و k قرار می‌دهیم و مراحل بالا را دوباره اجرا می‌کنیم. (اگر x دیگر رقمی نداشت کار را متوقف می کنیم و عدد شمارنده‌ی node را به جواب نهایی اضافه می‌کنیم. )

هر کدام از عملیات های بالا از O(lg(x)) هستند. زیرا تعداد مراحل اجرا شدن هرکدام دست بالا به اندازه‌ی تعداد رقم های عدد x در نمایش آن در مبنای 2 است و هر مرحله نیز تعداد ثابتی عملیات دارد. واضح است که با n بار استفاده کردن از عملیات های بالا ، با حافظه و زمانی از $O(n \times lg(\max_{01 \le i \le 1} a_i))$ می توان تعداد کل زوج های مورد نظر را پیدا کرد. (چرا؟)

اثبات درستی الگوریتم به عهده‌ی خودتان ☺

راه حل پرسش سوم " سرزمین ئمت باستان" :

محمدجواد توانست به راحتی تعداد بلورها را پیدا کند. او سه سوال زیر را پرسید.

یک: آیا تعداد بلورهای مورد نیاز یک یا دوتاست؟

دو: آیا تعداد بلورهای مورد نیاز یک یا سه‌تاست؟

سه: آیا تعدا بلورهای موردنیاز یک یا چهارتاست ؟

برای اثبات سه حالت زیر را در نظر می گیریم:

پاسخ «یک» باشد: در این صورت جواب حداقل دو تا از سوال ها باید بله باشد.

پاسخ «دو،سه،چهار» باشد: در این صورت گدلمت دروغ خودش را در سوال اول گفته بنابراین دقیقا به بک از سوال ها پاسخ «بلی» می‌دهد.

پاسخ «پنج» باشد: در این صورت گدلمت به هر سه سوال بالا جواب منفی می‌دهد.

واضح است که هیچکدام ازین مجموعه جواب‌ها با هم اشتراکی ندارند. بنابراین با هر حالت جواب دادن گدلمت می‌توان به طور یکتا تعدا بلورها را شناسایی کرد.

**نکته:** در واقع ما باید سوال‌هایی بپرسیم که حتی اگر گدلمت در جواب ما یک بار دروغ بگوید باز هم موفق به یافتن جواب شویم. اگر جواب ها را به صورت صفرو یک دسته بندی کنیم، باید سوال‌هایی پیدا کنیم که رشته‌ی جواب‌های درستی که به آن ها باید داده شود با تعویض یک بیت قابل تبدیل به رشته‌ای دیگر(که این رشته برای عددی دیگر است) نباشد (ممکن است در یک سوال دروغ بشنویم).این ایده‌ی مسئله‌ی معروفی است که می‌خواهیم بین یک تا n یک عدد حدس بزنیم و شخص مقابل می‌تواند حداکثر یک دروغ بگوید.

به عنوان یک مسئله‌ی پیکارجو می‌توانید به مسئله‌ی زیر فکر کنید:

> *فرض کنید در همین سوال سوان به ما اجازه‌ی پرسیدن حداکثر $n$ پرسش دیگر داده است. سقف تعداد بلور هایی که یک ماشین حرکت دائم می‌تواند داشته باشد، به طوری که ما همواره بتوانیم این تعداد را پیدا کنیم چندتاست؟ (در سوال قبلی این تعداد 5 تا بود)*