

## برج هانوی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

سه میله‌ی - میله‌ی مبدأ (A) ، میله‌ی کمکی (B) و میله‌ی مقصد (C) - و تعدادی دیسک در میله‌ی مبدأ داریم. هدف انتقال تمام دیسک‌ها از این میله به میله‌ی مقصد با رعایت دو شرط زیر است:

- در هر زمان فقط یک دیسک را می‌توان جابجا نمود.
- نباید در هیچ زمانی دیسکی بر روی دیسک با اندازه‌ی کوچکتر قرار بگیرد.

به طور حتم می‌توان با روش آزمون و خطا به نتیجه‌ی مطلوب رسید. اما هدف ما ارائه‌ی الگوریتمی برای انتقال دیسک‌ها با کمترین جابجایی ممکن است.

به عنوان مثال، اگر  $n=2$  باشد:

(1) دیسک 1 را به میله‌ی B منتقل می‌کنیم ( $A \rightarrow B$ ):

(2) دیسک 2 را به میله‌ی C منتقل می‌کنیم ( $A \rightarrow C$ ):

(3) دیسک 1 را به میله‌ی C منتقل می‌کنیم ( $B \rightarrow C$ ):

توجه داشته باشید که بر اساس قانون اول، نمی‌توان به غیر از بالاترین دیسک هر میله، به دیسک دیگری از آن دسترسی پیدا کرد.

برای اینکه بتوان از روش بازگشتی برای حل یک مسئله استفاده نمود، مسئله باید قابلیت خرد شدن به زیرمسئله‌هایی از همان نوع مسئله‌ی اصلی و اندازه‌ی کوچکتر را داشته باشد. این ویژگی در مورد مسئله‌ی برج هانوی صدق می‌کند.

ایده‌ی اصلی از آنجا ناشی می‌شود که برای جابجا کردن بزرگترین دیسک از میله‌ی A به میله‌ی C، ابتدا باید تمامی دیسک‌های کوچکتر به میله‌ی B منتقل شوند. پس از تمام شدن این مرحله، دیسک بزرگ را از میله‌ی

A به میله‌ی C منتقل کرده و مجدداً به کمک میله‌ی A تمامی دیسک‌های میله‌ی B را به میله‌ی C منتقل می‌کنیم. پس به طور خلاصه می‌توان گفت:

مرحله‌ی یک:  $n-1$  دیسک بالایی میله‌ی مبدأ با شرایط ذکر شده و به کمک میله‌ی C به میله‌ی B منتقل می‌شوند.

مرحله‌ی دو: بزرگترین دیسک از میله‌ی مبدأ به میله‌ی مقصد منتقل می‌شود.

مرحله‌ی سه:  $n-1$  دیسک میله‌ی B با کمک گرفتن از میله‌ی A به میله‌ی مقصد منتقل می‌شوند.

می‌بینیم که توانستیم عملیات جابجا کردن  $n$  دیسک را به دو عملیات مشابه ولی با اندازه‌ی کمتر و یک عملیات ساده تقسیم کنیم.

## ورودی

ورودی شامل عدد صحیح  $n$  که تعداد دیسک را مشخص می‌کند.

$$1 \leq n \leq 20$$

## خروجی

خروجی شامل خط‌هایی که به ترتیب نشان می‌دهد که از دیسکی از کدام میله برداشته و به کدام میله می‌رود.

## مثال

### ورودی نمونه ۱

2

### خروجی نمونه ۱

A --> B  
A --> C  
B --> C

ورودی نمونه ۲

3

خروجی نمونه ۲

A --> C  
A --> B  
C --> B  
A --> C  
B --> A  
B --> C  
A --> C