پروژهٔ پایانی درس طراحی الگوریتم‌ها

نگار موقتیان، ۹۸۳۱۰۶۲

۱.

مکعب‌های رنگی وزندار:

در این مساله ویژگی optimal substructure برقرار نیست زیرا ساختن بلندترین برج تا هر مرحله ساخت بلندترین برج در مراحل بعدی را تضمین نمی‌کند. همچنین نمی‌توان مساله را بخش‌های کوچک‌تری تقسیم کرد و بخش‌ها را به صورت جداگانه حل کرد زیرا رنگ سطح پایین برج بالایی به رنگ سطح بالای برج زیرین بستگی دارد. انتخاب حریصانه‌ای نیز وجود ندارد که ما را به طور قطعی به پاسخ مساله برساند. بنابراین به نظر می‌رسد باید این مساله را از طریق بررسی فضای حالات حل کنیم.

این کار را باید به صورت روشمندی انجام دهیم، لذا از روش ساخت درخت فضای حالات و شاخه و حرص استفاده می‌کنیم.

طبق شرط دوم صورت سوال مکعب‌هایی که وزن بیش‌تری دارند باید زیر مکعب‌هایی که وزن کمتری دارند قرار بگیرند. برای برآورده ساختن این شرط ابتدا مکعب‌ها را به ترتیب وزنشان و به طور نزولی مرتب می‌کنیم. حال در مورد هر مکعب به ترتیب تصمیم می‌گیریم.

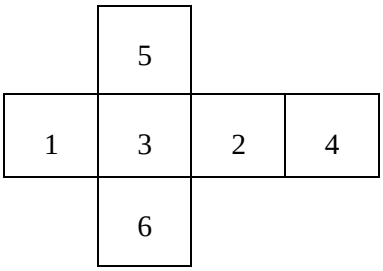
برای مکعب اول ۷ حالت قابل بررسی است (به شرط اینکه این مکعب رنگ تکراری نداشته باشد. در غیر این صورت تعداد حالات کمتر است زیرا در ادامه مساله تفاوتی ایجاد نمی‌شود که برای مثال کدام سطح قرمز به سمت بالا باشد). حالت اول این است که مکعب اول را در کل انتخاب نکنیم. حالت دوم این است که مکعب اول را انتخاب کنیم و وجه اول آن به سمت بالا باشد. حالت سوم این است که مکعب اول را انتخاب کنیم و وجه دوم آن به سمت بالا باشد و به همین ترتیب.

حال در هر یک از این ۷ شاخه برای مکعب دوم نیز تصمیم می‌گیریم. در شاخهٔ اول باز برای مکعب دوم ۷ حالت داریم زیرا هنوز هیچ مکعبی انتخاب نشده و محدودیتی نداریم. اما در شاخه‌های دیگر حالات کمتری داریم زیرا سطح زیرین مکعب دوم با سطح بالایی مکعب قبل باید همرنگ باشد. به همین ترتیب برای مکعب-های بعدی نیز ادامه می‌دهیم.

با همین روش کل درخت حالات را می‌سازیم تا برای مکعب n ام نیز تمام حالات بررسی شود. حال می‌توانیم برگ‌هایی که بیش‌ترین ارتفاع را دارند را به عنوان پاسخ مساله اعلام کنیم.

در حل این مساله دو موجودیت پیاده‌سازی شده‌اند:

۱. مکعب (Cube): هر مکعب یک وزن، آرایه‌ای از رنگ‌های هر وجه و یک اندیس دارد که نوبت آن در وارد کردن ورودی‌هاست. برای سادگی در پیاده سازی ترتیب رنگ‌های ورودی به صورت زیر می‌باشد:



۲. گره (Node): هر گره یک حالت از تصمیمات گرفته شده را نمایندگی می‌کند. یک رشته desc دارد که نمایانگر این است که در این مرحله چه تصمیمی گرفته شده. یک height دارد که ارتفاع برج تا این لحظه با توجه به تصمیمات گرفته شده‌است. یک maxHeight دارد که نشان‌دهندهٔ این است که با توجه به تصمیمات گرفته شده در بهترین حالت ارتفاع برج چقدر خواهد بود، از این مقدار می‌توانیم برای حرص کردن شاخه‌ها استفاده کنیم. یک topColor دارد که رنگی است که در این لحظه در سطح بالای برج قرار دارد. این مقدار در تصمیم‌گیری‌های بعدی حائز اهمیت است. یک parent دارد که نشان می‌دهد گره قبلی آن چه بوده. از این طریق می‌توانیم از هر برگ دلخواه به ریشه برسیم و این مساله در چاپ کردن پاسخ نهایی کاربرد دارد. در نهایت نیز یک آرایه به نام children دارد که بچه‌های این گره را مشخص می‌کند.

تحلیل زمانی الگوریتم:

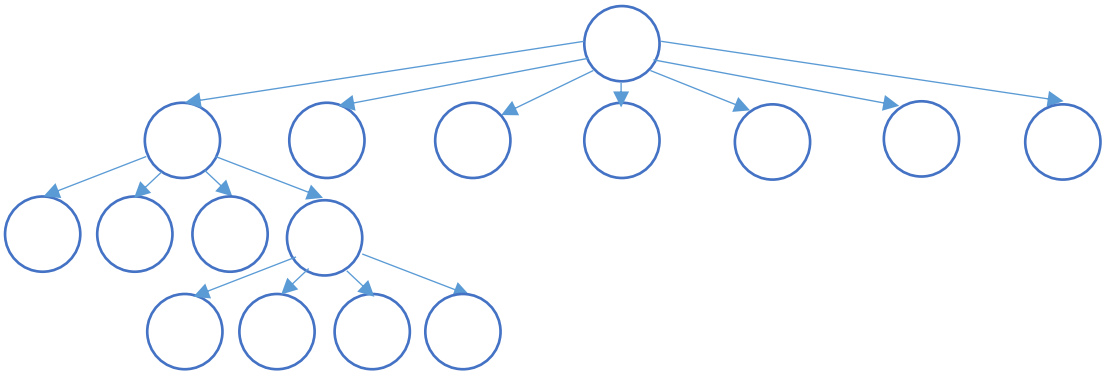
الگوریتم گفته شده از طریق بررسی فضای حالات به پاسخ مساله می‌رسد. می‌دانیم شرایط حرص ممکن است پیش نیاید پس در بدترین حالت هیچ حرصی نداریم و باید تمامی حالات را بررسی کنیم.

در ابتدا گرفتن ورودی‌ها به تعداد مکعب‌ها انجام می‌شود و زمان آن برابر است با O(n).

پس از آن باید n مکعب داده شده را sort کنیم. این کار در زمان O(nlogn) قابل انجام است.

پس از آن باید درخت فضای حالات را بسازیم. همانطور که گفته شد برای مکعب اول ۷ حالت داریم. برای مکعب‌های بعدی حداکثر ۳ وجه می‌تواند به طور یکتا رنگ مورد نظر را داشته باشد، یک حالت نیز برای انتخاب نکردن مکعب داریم، پس در کل هر گره می‌تواند ۴ فرزند داشته باشد.

بنابراین درخت حالات به شکل زیر می‌باشد:



به علاوه ارتفاع درخت با تعداد مکعب‌ها یعنی n برابر است. بنابراین تعداد کل گره‌ها برابر است با:

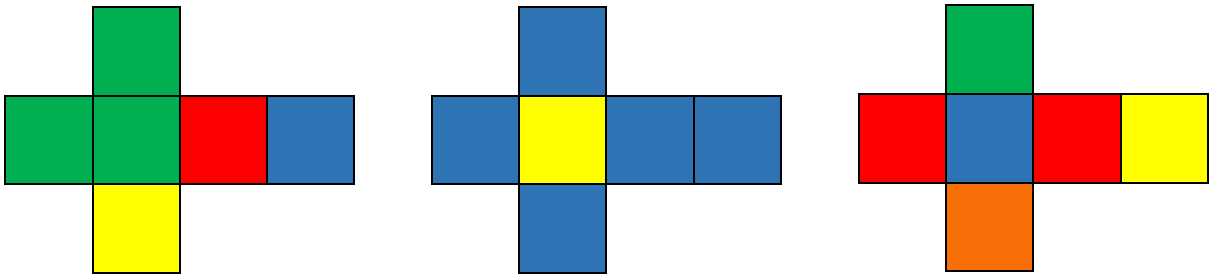
$$1+\sum_{i=1}^{n} 7\times 4^{i-1} = 1+\frac{7}{4}\sum_{i=1}^{n} 4^{i} = 1+\frac{7}{4}\frac{1-4^{n-1}}{1-4} = 1+\frac{7}{12}(4^{n-1}-1) \in O(4^n)$$

چاپ کردن پاسخ‌ها نیز نیاز به پیمایش دوبارهٔ همین درخت دارد و مرتبهٔ زمانی آن با مرتبهٔ زمانی مرحلهٔ قبل برابر است.

در نتیجه زمان اجرای این الگوریتم نمایی است.

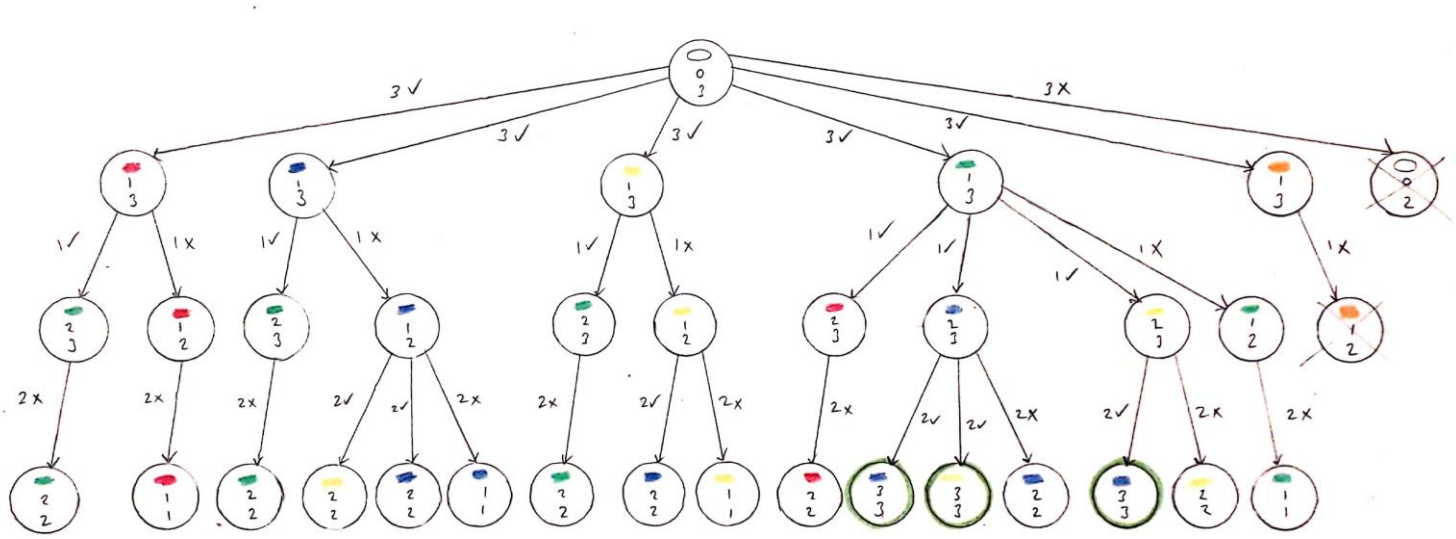
مثالی از این الگوریتم:

فرض کنید ۳ مکعب مانند زیر داریم.



Cube 1 – weight = 15 Cube 2 – weight = 5 Cube 3 – weight = 30

درخت حالات مانند شکل زیر خواهد شد. در هر گره به ترتیب رنگ سطح بالایی برج، ارتفاع برج و حداکثر ارتفاع قابل وصول آن نوشته شده است (ساخت درخت حالات مشابه DFS انجام می‌شود).



همانطور که دیده می‌شود ۳ تا از شاخه‌ها به برجی با ارتفاع ۳ می‌رسند. این سه پاسخ مساله هستند. حال اگر بخواهیم ببینیم طبق چه انتخاب‌هایی به این پاسخ رسیده‌ایم کافیست از برگ‌های پاسخ به سمت ریشه حرکت کنیم و انتخابی که کرده‌ایم را بخوانیم. برای این سه پاسخ از چپ به راست می‌توان گفت:

۱. مکعب ۲ را طوری قرار می‌دهیم که وجه آبی آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه آبی آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه سبز آن بالا باشد
۲. مکعب ۲ را طوری قرار می‌دهیم که وجه زرد آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه آبی آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه سبز آن بالا باشد
۳. مکعب ۲ را طوری قرار می‌دهیم که وجه آبی آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه زرد آن بالا باشد – مکعب ۱ را طوری قرار می‌دهیم که وجه سبز آن بالا باشد

(اولین مکعبی که نوشته شده در برج پاسخ بالاترین مکعب است)

نمونهٔ ورودی و خروجی برنامه:

شکل زیر نمونهٔ ورودی و خروجی برنامه مطابق با مثال بالا می‌باشد:

```
Number of cubes: 3
The weight of the cube number 1: 15
The colors of the cube number 1: g r g b g y
The weight of the cube number 2: 5
The colors of the cube number 2: b b y b b b
The weight of the cube number 3: 30
The colors of the cube number 3: r r b y g o

MAXIMUM HEIGHT: 3

Solution number 1:
    cube 2 | top color: b
    cube 1 | top color: b
    cube 3 | top color: g

Solution number 2:
    cube 2 | top color: y
    cube 1 | top color: b
    cube 3 | top color: g

Solution number 3:
    cube 2 | top color: b
    cube 1 | top color: y
    cube 3 | top color: g

Process finished with exit code 0
```