بسمه تعالی

نمونه سوالات

برنامه نویسی پویا

طراحی الگوریتم

نوروز 1395

سوال ۱:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n) که n مقدار عدد ورودی است، حل کنید.

جمشید راننده تاکسی است. او همیشه از اینکه مسافرین پول‌خرد ندارند، گله مند است! او اعتقاد دارد که اگر فقط پول های 3، 5 و 7 ریالی داشتیم، می توانستیم تعداد زیادی از مقادیر را با این سه نوع پول بسازیم و اعتقاد دارد دولت باید سریعا، پول ها را جمع کند و فقط این سه نوع پول را بین مردم پخش کند! او از شما می خواهد برنامه ای بنویسید که به بررسی درستی این مساله بپردازد.

او از شما می خواهد که برنامه ای بنویسید که ورودی آن، یک عدد صحیح باشد؛ مقدار پولی که می خواهیم با 3 ریالی، 5 ریالی و 7 ریالی بسازیم. و در صورتی که امکان ساختن آن عدد وجود داشت در خروجی YES و در غیر این صورت NO را چاپ کنید.

ورودی نمونه: 10
خروجی نمونه: YES

در این مثال با توجه به اینکه با یک 3 ریالی و یک 7 ریالی می توانیم 10 ریال را بسازیم، جواب مساله مثبت است.

ورودی نمونه: 1
خروجی نمونه: NO

قطعا نمی توانیم عدد 1 را با این سه نوع پول بسازیم!

ورودی نمونه: 6
خروجی نمونه: YES

شما می توانید با دو تا 3 ریالی، 6 ریال را بسازید.

ورودی نمونه: 32
خروجی نمونه: YES

$$32 = 3 + 7 + 11 + 11 \rightarrow YES$$

سوال ۲:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n) که n طول تابلو است، حل کنید.

در یک مراسم سال نو، قصد داریم با کاشی های رنگی، یک شکل زیبا درست کنیم. ما کاشی قرمز با ابعاد3×1 و کاشی سبز با ابعاد 5×1 و کاشی آبی فیروزه ای با ابعاد 7×1 است. و می خواهیم آنها را در یک تابلو با ابعاد $n \times 1$ بچینیم. به طوری که کل ردیف را بپوشاند. و کاشی ای بیرون نزند. برنامه ای بنویسید که تعداد حالات ممکن برای این کار را بشمارد.

ورودی برنامه مقدار n است. خروجی برنامه تعداد حالات ممکن برای تزیین این تابلو است.

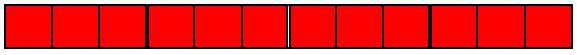
ورودی نمونه:
8
خروجی نمونه:
2

در این مثال تابلوی 8×1 می‌توانیم به دو صورت با کاشی ها تزیین کنیم.

حالت اول:

حالت دوم:

ورودی نمونه:
12
خروجی نمونه:
3



همانطور که می بینید، سه حالت وجود دارد که بتوانیم تابلو 12×1 را با کاشی های خود تزیین کنیم.

ورودی نمونه:
5
خروجی نمونه:
1

ورودی نمونه:
4
خروجی نمونه:
0

ورودی نمونه:
40
خروجی نمونه:
2643

سوال ۳:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n) که n مقدار عدد ورودی است، حل کنید.

جمشید که احساس می‌کند این تغییر وی در پول می‌تواند باعث رشد اقتصادی زیادی شود و از شما می خواهد برنامه ای بنویسید که دقیقا مشخص کند به چند روش می‌توانیم یک مقدار پول را با 3، 5 و 7 ریالی بسازیم.

برنامه ای بنویسید که یک عدد صحیح را به عنوان ورودی بگیرد و مشخص کند که به چند طریق می‌توان آن عدد را با جمع اعداد 3، 5 و 7 ساخت.

توجه کنید که 5+5+7 با 5+7+5 تفاوتی ندارد و یکسان هستند.

ورودی نمونه:
17
خروجی نمونه:
3

$$17 = 7 + 7 + 3 = 7 + 5 + 5 = 5 + 3 + 3 + 3 + 3 \rightarrow answer = 3$$

ورودی نمونه:
12
خروجی نمونه:
2

$$12 = 7 + 5 = 3 + 3 + 3 + 3 \rightarrow answer = 2$$

دقت کنید که در این سوال حالت 7+5 با حالت 5+7 یکسان در نظر گرفته می شوند.

ورودی نمونه:
7
خروجی نمونه:
1

ورودی نمونه:
4
خروجی نمونه:
0

عدد 4 را نمی‌توان ساخت.

ورودی نمونه:
40
خروجی نمونه:
11

11 =

5 + 7 + 7 + 7 + 7 + 7

5 + 5 + 5 + 5 + 5 + 5 + 5 + 5

3 + 5 + 5 + 5 + 5 + 5 + 5 + 7

3 + 3 + 5 + 5 + 5 + 5 + 7 + 7

3 + 3 + 3 + 5 + 5 + 7 + 7 + 7

3 + 3 + 3 + 3 + 7 + 7 + 7 + 7

3 + 3 + 3 + 3 + 3 + 5 + 5 + 5 + 5 + 5

3 + 3 + 3 + 3 + 3 + 3 + 5 + 5 + 5 + 7

3 + 3 + 3 + 3 + 3 + 3 + 3 + 5 + 7 + 7

3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 5 + 5

3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 7

سوال ۴:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n*maxWeight) که n تعداد کالا ها است، حل کنید.

فروشنده دوره گردی داریم که یک کوله پشتی دارد و می خواهد اجناس خود برای فروش را در کوله پشتی قرار دهد. و سپس در شهر های مختلف برود و اجناس خود را بفروشد. او بسیار علاقه مند است که تمام وسایل خود را در کوله پشتی خود بگذارد و ببرد. اما کوله پشتی‌اش از لحاظ تحمل وزن ظرفیت محدودی دارد و نمی‌تواند وزن بیشتر از maxWeight را تحمل کند.

فروشنده دوره‌گرد لیست قیمت اجناس و وزن آنها را به شما می‌دهد.

او از شما می خواهد برنامه ای بنویسید که مشخص کند کدام یک از کالا ها را داخل کوله پشتی بگذارد و کدام یک را نگذارد تا ارزش کالاهای داخل کوله پشتی حداکثر شود.

در خط اول ورودی، maxWeight داده می‌شود.

در خط بعدی تعداد اجناس n.

و در n خط بعدی، ارزش و وزن اجناس به ترتیب مشخص می‌شود. ارزش و وزن اجناس اعداد صحیح هستند.

خروجی برنامه دارای n خط است. در صورتی که در حالت بهینه، کالای iام برداشته شده است؛ در خط i ام YES و در غیر این صورت NO چاپ کنید.

ورودی نمونه:
10 3 4 9 2 4 3 4
خروجی نمونه:
NO YES YES

در این مثال ما سه کالا با ارزش 2 و 3 و 4 داریم. در صورتی که کالای با ارزش 4 را برداریم، دیگر کالا های دیگر را نمی‌توانیم برداریم زیرا از وزن آن از 10 بیشتر می‌شود. در صورتی که کالای با ارزش 3 را برداریم؛ می‌توانیم کالای با ارزش 2 را نیز برداریم و در کل، جمع ارزش اجناس داخل کوله پشتی ما 5 می‌شود.

ورودی نمونه: 17 3 4 9 2 4 3 4
خروجی نمونه: YES YES YES

در این حالت با توجه به اینکه ظرفیت کوله پشتی ما بیشتر شده است، می توانیم تمام اجناس را در کوله پشتی قرار دهیم در نتیجه جمع ارزش اجناس داخل کوله پشتی حداکثر می‌تواند 9 باشد.

سوال ۵:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(m*n) که n و m ابعاد باغ هستند، حل کنید.

دو نفر در یک باغ شریک هستند. نقشه باغ آن ها به صورت یک ماتریس داده شده است. آنها می خواهند باغ را به دو قسمت تقسیم کنند. به طوری که این تقسیم، عادلانه باشد. عدد داخل هر خانه از ماتریس نشان می‌دهد که آن درخت چه میزان سود برای صاحبش دارد. ممکن است یک درخت میوه ندهد و فقط خرج آب و مراقبت داشته باشد. درنتیجه ممکن است مقدار بعضی از خانه های ماتریس منفی نیز باشد.

منظور از عادلانه این است که بعد از تقسیم، اختلاف بین سود طرف چپ و طرف راست حداقل باشد.

این دو نفر می خواهند باغ را به صورت عمودی تقسیم کنند.

برای مثال:

0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	2	0	0	0

باغ به دو قسمت تقسیم شده است. و سود کل طرف آبی برابر با 2 است و سود کل طرف سبز برابر با 1. و در این حالت اختلاف سود طرف چپ و راست کمترین است.

در خط اول ورودی دو عدد n و m داده میشود. که ابعاد باغ را نشان میدهد.

سپس در n خط بعدی، در هرخط m عدد داده می‌شود که نشان دهنده مقدار سود خانه ها هستند.

در خروجی تعداد ستون های متعلق به بخش سمت چپ را چاپ کنید.

ورودی نمونه: 4 7 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0
خروجی نمونه: 4

در این مثال، خروجی 4 یعنی تعداد ستون های قسمت تقسیم شده سمت چپ، 4 تا است.

دقت کنید که تقسیم ما لزوما باید عمودی باشد. یعنی باغ را به دو قسمت، سمت چپ و سمت راست تقسیم کنید. مانند شکل زیر:

0	0	0	0	0	0	0
0	0	0	0	1	0	-1
0	0	0	0	0	0	-1


-2
0
0
2
0
0
0

سوال ۶:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O((m*n)*(sum)) که n و m ابعاد باغ هستند و sum جمع سود این کل درختان باغ است، حل کنید.

این دو شریک باغ در مساله قبل، تصمیم می گیرند برای اینکه عدالت بیشتری برقرار باشد باغ را به روشی دیگر بین خود تقسیم کنند. فرض کنید که مانند سوال قبل، نقشه باغ به صورت ماتریس داده شده است. و در هر خانه از این ماتریس، فقط یک درخت وجود داشته باشد. آنها تصمیم می‌گیرند که مشخص کنند هر کدام از درختان مربوط به کدام یک از آنهاست. به این صورت که درخت ها را با دو رنگ، رنگ‌آمیزی کنند. و بصورتی که اختلاف سود درخت های این دو شخص، کمترین مقدار ممکن شود.

در این مساله هیچکدام از خانه های آرایه منفی نیستند.

برای مثال:

0	0	0	4	0	0	0
0	0	0	4	0	0	0
0	0	0	4	0	0	0
0	0	0	4	0	0	0

در این حالت، خانه های آبی برای نفر اول و خانه های سبز متعلق به نفر دوم هستند.

درختان به گونه ای رنگ شده اند که که اختلاف مجموع اعداد رنگ آبی با مجموع اعداد رنگ سبز، کمترین باشد.

در خط اول ورودی دو عدد n و m داده میشود. که ابعاد باغ را نشان میدهد.

سپس در n خط بعدی، در هرخط m عدد داده می‌شود که نشان دهنده مقدار سود خانه ها هستند.

در خروجی مساله شما باید به ازای تمام نقاط نقشه عدد 1 و یا 2 را چاپ کنید. 1 به معنی این است که آن خانه متعلق به نفر اول است و 2 به معنی اینکه این خانه متعلق به نفر دوم است. جواب شما باید طوری باشد که اختلاف سود بین نفر اول و دوم حداقل شود.

ورودی نمونه:

```
4 7
0 0 0 4 0 0 0
0 0 0 4 0 0 0
0 0 0 4 0 0 0
0 0 0 4 0 0 0
```

خروجی نمونه:

```
1 1 1 1 1 1 1
1 1 1 2 1 1 1
1 1 1 2 1 1 1
```

1 1 1 1 1 1 1

ورودی نمونه:

4 7
0 0 0 0 0 1 0
0 0 0 0 2 0 0
0 0 0 3 0 0 0
0 0 4 0 0 0 0

خروجی نمونه:

1 2 1 1 1 1 1
1 2 2 1 2 1 1
1 1 1 2 1 1 1
1 1 1 1 1 1 1

سوال ۷:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n*n*m) که n و m ابعاد باغ هستند، حل کنید.

بعد از چند ماهی که از تقسیم بندی باغ می‌گذرد، بعلت اینکه درخت ها با رنگ های دارای سرب، رنگ شده بودند، تعداد زیادی از درختان آسیب دیده و داشتن آنها سودی برای این دو شریک ندارد. و در نتیجه این دو شریک تصمیم می گیرند قسمتی از باغ خود را بفروشند. به طوری که قسمتی از باغ که برایشان می ماند یک مستطیل است که داشتن آن بیشترین سود ممکن را حاصل می کند.

چهار گوشه این مستطیل مختصات

$$(n1, m1), (n1, m2), (n2, m1), (n2, m2)$$

را دارند که:

$$0 \leq m1 \leq m2 \leq m$$

$$0 \leq n1 \leq n2 \leq n$$

برای مثال در باغی به شکل زیر:

-1	-1	-1	-1	-1	-1	-1
-1	-1	2	2	2	-1	-1
-1	-1	2	-1	2	-1	-1
-1	-1	2	2	2	-1	-1
-1	-1	-1	-1	-1	-1	-1

در این مساله، مستطیل نهایی که بیشترین سود ممکن را دارد، مستطیل سبز است.

ورودی مانند مساله قبل، نقشه باغ است.

خروجی به ترتیب، n1، n2، m1، m2 است.

ورودی نمونه: 5 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 2 2 2 -1 -1 -1 -1 2 -1 2 -1 -1 -1 -1 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1
خروجی نمونه:

1 3 3 5

سوال ۸:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n*n) که n طول آرایه است، حل کنید.

اگر یک دنباله داشته باشیم، یک زیردنباله یک زیر مجموعه از آن دنباله است. به طوری که ترتیب اعضای دنباله عوض نشده باشد.

برای مثال دنباله‌ی

0, 8, 4, 12, 2, 10, 6, 14

را در نظر بگیرید.

برای مثال دنباله‌ی

8,12,2,14

یک زیر دنباله از دنباله اولیه است.

برنامه ای بنویسید که به ازای دنباله ورودی، بزرگترین زیر دنباله‌ی آن که غیر نزولی است را بدهد.

ورودی مساله یک آرایه از اعداد است. خروجی بزرگترین زیردنباله از آن دنباله است که غیر نزولی باشد.

ورودی نمونه: 16 0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15
خروجی نمونه: 0 2 6 9 11 15

در صورتی که چندین جواب وجود داشت می توانید به دلخواه، هرکدام را که خواستید چاپ کنید.

سوال ۹:

در مساله سوم تمرین سری 5، گفتیم که حق نداریم جعبه ها را بچرخانیم.

حال اگر این امکان وجود داشته باشد که فقط در محور z 90 درجه جعبه ها را بچرخانیم، برای مثال

جعبه با طول 10 و عرض 8 و ارتفاع 1 را با این چرخش به جعبه ای با طول 8 و عرض 10 و ارتفاع 1 می‌توان تبدیل کرد.

در واقع اگر فقط بتوان جای طول و عرض جعبه‌ها را عوض کرد؛ در آن صورت مساله را چگونه باید حل کرد؟

ورودی نمونه: 3 8 10 8 10 10 8
خروجی نمونه: 3

سوال ۱۰:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n*m) که n طول رشته a و m طول رشته b است، حل کنید.

رشته a به طول n و رشته b به طول m را داریم. می خواهیم با کمترین هزینه رشته b را تبدیل به رشته a بکنیم.

کار های مجاز: حذف یه حرف از b به هزینه c1. حذف یک حرف از a به هزینه c2. تغییر یک حرف از b به هزینه c3. تغییر یک حرف از a به هزینه c4.

ورودی مساله رشته a و رشته b و سپس به ترتیب هزینه های c1، c2، c3، c4 .

خروجی حداقل هزینه برای تبدیل رشته b به رشته a است.

ورودی نمونه: aaa bbb 1 1 3 3
خروجی نمونه: 6

با حذف تمام حروف a و تمام حروف b، این دو رشته خالی مثل هم می شوند.

ورودی نمونه: xyz mno 3 3 1 1
خروجی نمونه: 3

می توان تمام حروف رشته a را تغییر داد. z→o، y→n،x→m . سه تغییر که هزینه هر کدام 1 است. در نهایت هزینه 3 خواهد بود.

ورودی نمونه: xyz mno 10 10 1 2
خروجی نمونه: 3

می توان تمام حروف رشته b را تغییر داد. سه تغییر که هزینه هر کدام 1 است. در نهایت هزینه 3 خواهد بود. در صورتی که حروف رشته a را تغییر دهیم هزینه بیشتری برایمان خواهد داشت.

ورودی نمونه:

xxx
xxy
1 1 5 5

خروجی نمونه:

2

در صورتی که y را به x تغییر بدهیم، هزینه 5 داریم. و در صورتی که حرف x از رشته a و y از رشته b را حذف کنیم هزینه 2 خواهیم داشت.

سوال ۱۱:

فرض کنید تابعی داریم به شکل زیر:

```
int q12(string a, string b, int c1, int c2, int c3, int c4)
```

که سوال 12 را حل میکند. (یعنی به ازای ورودی های سوال 12، خروجی های مطلوب مساله 12 را برمیگرداند.

حال با استفاده از این تابع، سوال 13 را حل کنید.

ورودی مساله به ترتیب رشته a و رشته b و سپس c1، c2، c3، c4، c5، c6 .

حذف یک حرف از b به هزینه c1. حذف یک حرف از a به هزینه c2. تغییر یک حرف از b به هزینه c3. تغییر یک حرف از a به هزینه c4. اضافه کردن یک حرف به b به هزینه c5. اضافه کردن یک حرف به a به هزینه c6.

خروجی مساله: حداقل هزینه ممکن برای تبدیل رشته b به رشته a.

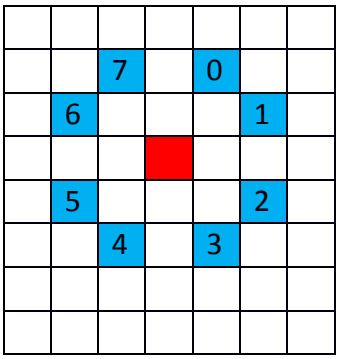
ورودی نمونه برای مساله 13: xxx xxy 5 2 10 12 6 1
تبدیل میشود به: ورودی نمونه برای مساله 12: xxx xxy 1 2 10 12
خروجی نمونه: 3

در واقع با اضافه کردن حرف y به رشته a به هزینه 1 و حذف حرف x از رشته a به هزینه 2، در نهایت با هزینه 3، رشته b تبدیل به رشته a شده است.

سوال ۱۲:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(d*d*n) که n تعداد حرکت و d طول صفحه شطرنج است، حل کنید.

یک صفحه شطرنج داریم. یک مهره اسب داریم که به صورت L شکل حرکت می‌کند. به این صورت که اگر اسب در نقطه قرمز باشد در یک حرکت می‌تواند به هرکدام از نقاط آبی برود.



می‌خواهیم بدانیم اسب با شروع از نقطه a به چند روش می‌تواند به نقطه b برود به شرطی که دقیقا n حرکت کرده باشد.

توجه کنید که اشکالی ندارد از یک نقطه چند بار عبور کرده باشیم.

اگر مسیر را به صورت رشته ای از اعداد 0 تا 7 نشان دهیم (هر عدد نشان‌دهنده نوع حرکت در آن مرحله است)، در صورتی که رشته معادل دو مسیر متفاوت باشد، آن دو مسیر متفاوت محسوب می‌شوند.

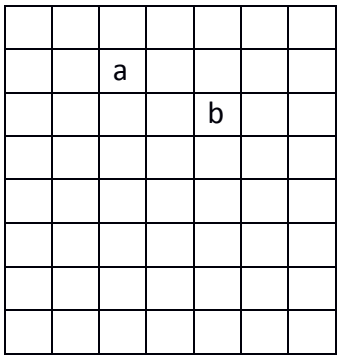
با توجه به اینکه ممکن است جواب مساله مقدار زیادی شود، باقیمانده آن به 1000000007 را در خروجی چاپ کنید.

یعنی اگر 1000000008 تا روش وجود داشت شما 1 را در خروجی چاپ کنید.

ورودی مساله، طول صفحه شطرنج (صفحه شطرنج مربعی است) و سپس نقاط a و b می باشد. و سپس n که تعداد حرکات است.

ورودی نمونه:
8 2 3 3 5 1
خروجی نمونه:
1

در این مثال، تنها به یک روش می توان از a به b رفت.



ورودی نمونه:
8 4 4 4 4 2
خروجی نمونه: 8

ورودی نمونه:
8 4 4 4 4 6
خروجی نمونه: 4370

سوال ۱۳:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(d*d*n) که n تعداد حرکت و d طول صفحه شطرنج است، حل کنید.

این مساله را با مرتبه حافظه ای O(d*d) حل کنید.

در مساله 14 تعداد راه های رسیدن از a به b با دقیقا n حرکت خواسته شده بود.

در مساله 15 تعداد راه های رسیدن از a به b با حداکثر n حرکت خواسته شده است.

ورودی نمونه: 8 4 4 4 4 2
خروجی نمونه: 9

دقت کنید اینکه با هیچ حرکتی از a به b رسیدن یک جواب قابل قبول است.

ورودی نمونه: 8 1 1 2 3 4
خروجی نمونه: 9

سوال ۱۴:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(d*d*k) که k تعداد ارقام است و d تعداد رقم های ما یعنی 10 است، حل کنید.

پژمان اعتقاد دارد که بعضی اعداد جادویی هستند. پژمان اعداد جادویی را دوست دارد و تصمیم دارد یک کلکسیون از آنها را جمع آوری کند. و به تابلویی در اتاقش بچسباند. او می خواهد بداند که تعداد اعداد جادویی k رقمی چه تعداد است. به همین دلیل از شما خواسته است برنامه ای بنویسید که مشخص کند تعداد اعداد جادویی k رقمی چه تعداد است.

اعداد جادویی ویژگی خاصی دارند. ویژگی آنها این است که بعضی از اعداد خاص دو رقمی در آنها دیده میشود.

برای مثال تمام اعدادی که در آنها 13 دیده میشود جادویی هستند. برای مثال عدد 551341 یک عدد جادویی است. اما عدد 31444 جادویی نیست.

در ورودی مساله اعدادی دو رقمی ای که وجودشان در عدد باعث جادویی شدن میشود داده می شود.

شما باید بگویید تعداد اعداد جادویی چقدر است.

با توجه به اینکه این عدد ممکن است زیاد باشد، شما باقیمانده آن را به 1000000007 بدست آورید.

خط اول ورودی k تعداد رقم ها است.

در خط بعد n که تعداد اعداد دو رقمی جادویی است.

در n خط بعد n عدد دو رقمی داده می شود.

در خروجی باقیمانده تعداد اعداد جادویی به 1000000007 را چاپ کنید.

راهنمایی: برای محاسبه این مقدار، می توانید اعداد غیر جادویی را حساب کنید و سپس از (k-1^10) آن را کم کنید.

همچنین می‌توانید به صورت مستقیم نیز حساب کنید. عدد جادویی ما، به ازای اینکه i رقم راست عدد را در نظر بگیریم و بقیه ارقام را نگاه نکنیم؛ می‌تواند جادویی باشد و یا می‌تواند جادویی نباشد. می‌توان اعداد جادویی را به این صورت تقسیم به حالت های مختلف تبدیل کنید. به ازای هر عدد جادویی یک i وجود دارد که اگر i رقم راست عدد را در نظر بگیریم، آن عدد جادویی نیست ولی به ازای i+1 عدد سمت راست، این عدد جادویی است.

ورودی نمونه: 2 3 13 17 19
خروجی نمونه:

3

اعداد 13، 17 و 19 جادویی هستند.

ورودی نمونه: 1 4 13 17 71 15
خروجی نمونه: 0

در تمام اعداد 1 تا 9، نمی‌توان عدد های 13، 17، 71، 15 را دید. پس تمام آنها غیر جادویی هستند.

ورودی نمونه: 3 1 12
خروجی نمونه: 19

120, 121, … , 129, 112, 212, 312, …912 → 19

سوال ۱۵:

این مساله را به روش برنامه نویسی پویا و با مرتبه زمانی O(n*(2^n)) که n تعداد شهر ها است، حل کنید.

فروشنده دورگرد ما که در سوالات قبلی کوله پشتی خود را با کالا های مناسب پر کرده بود، هم اکنون می‌خواهد راهی سفر شود. او می‌خواهد به n شهر سفر کند و محصولاتش را بفروشند. اما نمیداند که به چه ترتیبی. او می‌خواهد از هر شهر فقط یک بار عبور کند و کل مسیری که می‌پیماید حداقل شود. فاصله شهر ها در ماتریسی به ما داده شده است. با توجه به آن طول کل مسیر را می توان بدست آورد.

در حالت کلی، او باید (n-1)! حالت را بررسی کند. (او ابتدا در شهر 1 است.) اما این کار بسیار طولانی است. به خاطر همین او از ما می‌خواهد برنامه ای بنویسیم که با مرتبه زمانی n*(2^n) حل کند.

در خط اول ورودی n تعداد شهر ها است. در n خط بعد ماتریس مجاورت فاصله بین شهر ها داده شده است.

در خروجی، رئوس کوتاه ترین مسیر را چاپ کنید.

برای نشان دادن مسیر، باید رئوسی که فروشنده طی می‌کند به ترتیب چاپ شود. اما هر دور می‌تواند به n حالت نشان داده شود. بسته به آنکه اولین راس را چه در نظر بگیریم.

شما یکی از جواب ها را چاپ کنید.

ورودی نمونه:
4 0 10 1 10 1 0 10 10 10 10 0 1 10 1 10 0
خروجی نمونه:
31023

سوال ۱۶:

چنگیز (یکی از مربیان مهدکودک) به تازگی متوجه شده است که بچه ها در کلاس شادابی و نشاط همیشگی خود را ندارند. او از کودکان پرسید که چرا در کلاس شاداب نیستند. آنها گفتند که دوست دارند در کلاس بغل دست دوستانشان باشند. چنگیز تصمیم گرفت برای آنکه کیفیت کلاس بالا رود جای آنها به بهترین نحو تعیین کند.

در کلاس مهدکودک چنگیز، صندلی ها به صورت دوری چیده شده اند.

او ماتریس مجاورت میزان صمیمیت افراد را دارد که میزان صمیمیت افراد را با اعداد صحیح نشان داده است. او میداند هر نفر با دیگران چه میزان صمیمی است. او دوست دارد افرادی که با هم صمیمی هستند کنار هم بنشینند.

خوشحالی هر فرد بسته به اینکه سمت چپ و سمت راستش چه کسی بنشیند قابل محاسبه است. اگر با بغل دستی سمت راستش به میزان 7 صمیمی باشد و با بغل دستی سمت چپش به اندازه 2 در کل به اندازه 9 واحد خوشحال است.

چنگیز می‌خواهد طوری کلاس را بچیند که جمع خوشحالی افراد بیشینه شود.

فرض کنید که یک تابع داریم که طولانی ترین دور را برای فروشنده دوره گرد پیدا می‌کند. (برعکس سوال قبل که کوتاه ترین دور را پیدا می‌کرد! طولانی ترین دور را پیدا می‌کند)

برنامه ای بنویسید که با استفاده از این تابع، بهترین ترتیب را بدهد.

به عبارت دیگر، ورودی مناسبی به تابع دهید و با استفاده از این تابع خروجی مناسب را بدهید.

ورودی نمونه:

```
4
0 1 1 0
1 0 0 1
1 0 0 1
```

0 1 1 0
خروجی نمونه: 31023