

کوئیز ۴ - برنامه‌ریزی پویا

دقیقه 18 pm 5, 2021 در 12:19 - قابل دسترسی ژوئن 5, 2021 در 12:01 پرسش‌ها 5 امتیازها 13 مهلت ژوئن 5, 2021 در 12:19 محدودیت زمانی 16 دقیقه

Instructions

نکات مهم.

برای پاسخ، فقط یکبار می‌توانید به صفحه این کوئیز وارد شوید. امکان تلاش دوباره وجود ندارد.

هر سؤال فقط یکبار نمایش داده می‌شود. فقط قبل از رفتن به سؤال بعدی فرصت دارید جواب خود را تغییر دهید.

* پس از پاسخ و رفتن به سؤال بعدی امکان بازگشت و تغییر جواب را ندارید.

** مدت زمان پیشنهادی برای پاسخگویی به هر سؤال در بالای شرح سؤال آمده است ولی توجه داشته باشید که نهایتاً، مدیریت زمان پاسخگویی در کل بر عهده خود شماست.

*** به محض اینکه زمان‌سنج به یک دقیقه پایانی کوئیز وارد شد - چه پرسش آخر را پاسخ داده‌اید چه هنوز به آن نرسیده‌اید - حتماً خودتان با زدن دکمه [submit quiz] کوئیز را خاتمه دهید.

نمره	زمان	تلاش	تاریخچه تلاش‌ها
از 13 6.17	دقیقه 14	تلاش 1	آخرین

نمره این آزمون: 6.17 از 13

ارسال شد pm 5, 2021 در 12:15

این تلاش 14 دقیقه طول کشید

پرسش 1

امتیاز 2 / 1.5

** مدت زمان پیشنهادی برای پاسخگویی: ۳ دقیقه

-- برای حل مسئله P می‌خواهیم از روش برنامه‌ریزی پویا، استفاده کنیم.
برای این منظور، محاسبه مقدار بهینه مسئله اصلی را به محاسبه مقدار بهینه K تا زیرمسئله ربط می‌دهیم که هر زیرمسئله بر روی m پارامتر طرح می‌شود.
<<< بر این اساس، تناظر درست را بسازید:

پاسخ دادید

مسئله کوله‌پشتی صفر و یک

k=n-1, m=2

پاسخ درست

k=2, m=2

ادرست است

مسئله پرانتزبندی بهینه ضرب زنجیره‌ای ماتریس‌ها

k=n-1, m=2

ادرست است

مسئله برش میله

k=n-1, m=1

ادرست است

مسئله بلندترین زیرتوالی مشترک

k=2, m=2

Other Incorrect Match Options:

- k=3, m=2
- k=n-1, m=3

پرسش 2

امتیاز 2 / 0.5

** مدت زمان پیشنهادی برای پاسخگویی: ۳ دقیقه

-- با استفاده از روش برنامه‌ریزی پویا می‌خواهیم فقط ارزش جواب بهینه را محاسبه کنیم.
<<< بهترین الگوریتمی که از نظر هزینه فضای حافظه می‌شناسید را در نظر بگیرید و تناظر درست را بسازید.

-- T را نشان‌دهنده هزینه زمانی و S را نشان‌دهنده هزینه فضای حافظه در نظر بگیرید.

-- طول رشته‌ها در مسئله LCS را برابر فرض کنید.

پاسخ دادید

مسئله پرانتزبندی بهینه ضرب زنجیره‌ای ماتریس‌ها

 $T = O(n^3), S = O(n^2)$

پاسخ درست

 $T = O(n^3), S = O(n)$

پاسخ دادید

مسئله بلندترین زیرتوالی مشترک

 $T = O(n^2), S = O(n^2)$

پاسخ درست

 $T = O(n^2), S = O(n)$

ادرست است

مسئله برش بهینه میله

 $T = O(n^2), S = O(n)$

پاسخ دادید

مسئله کوله‌پشتی صفر و یک

 $T = O(n^2), S = O(n^2)$

پاسخ درست

 $T = O(n^2), S = O(n)$

Other Incorrect Match Options:

- $T = O(n), S = O(1)$
- $T = O(n^2), S = O(1)$
- $T = O(n^2), S = O(n^2)$
- $T = O(n^3), S = O(n^2)$

پرسش 3

امتیاز 2 / 1.33

** مدت زمان پیشنهادی برای پاسخگویی: ۲ دقیقه

مسئله پرانتزبندی بهینه برای ضرب زنجیره‌ای ماتریس‌ها را در نظر بگیرید:

الف) هزینه زمانی الگوریتم برنامه‌ریزی پویا برای این مسئله از مرتبه ([انتخاب]) است.ب) هزینه زمانی محاسبه تعداد ضرب‌های اسکالر به ازای پرانتزبندی بهینه در یک پیاده‌سازی بالا به پایین، بدون استفاده از به حافظه‌سازی از مرتبه ([انتخاب]) است.ج) هزینه زمانی چاپ بهترین پرانتزبندی از مرتبه ([انتخاب]) است.

پاسخ 1:

 n^3

ادرست است

پاسخ 2:

 n^2

پاسخ درست

 $\text{pow}(2, n)$

پاسخ درست

 n^3

پاسخ دادید

پاسخ 3:

 n

ادرست است

پرسش 4

امتیاز 3 / 1.5

** مدت زمان پیشنهادی برای پاسخگویی: ۳ دقیقه

-- شبه‌کد زیر، یک پیاده‌سازی از الگوریتم برنامه‌ریزی پویا برای حل مسئله بلندترین زیرتوالی مشترک است:

```
LCSZ( X, Y )
m ← X.length
n ← Y.length
Let Z[ 0..m, 0..n ] be a new table
for s = 0 to m+n do
  if s ≤ m then Z[ s, 0 ] ← 0
  if s-m ≤ n then Z[ 0, s-m ] ← 0
  for i = {{?A}} to {{?B}} do
    j ← s - i
    if {{?C}} then
      Z[ i, j ] ← Z[ i-1, j-1 ] + 1
    elseif Z[ i-1, j ] ≥ Z[ i, j-1 ] then
      Z[ i, j ] ← Z[ i-1, j ]
    else
      Z[ i, j ] ← Z[ i, j-1 ]
  return {{?D}}
```

<<< جاهایی که با علامت سؤال نشان داده شده‌اند را کامل کنید:

- {{?A}}
- {{?B}}
- {{?C}}
- {{?D}}

پاسخ 1:

پاسخ درست

$\max(1, s-n)$

پاسخ دادید

1

پاسخ 2:

پاسخ درست

$\min(m, s+1)$

پاسخ دادید

m

پاسخ 3:

ادرست است

$X[i] == Y[j]$

پاسخ 4:

ادرست است

$Z[m, n]$

پرسش 5

امتیاز 4 / 1.33

** مدت زمان پیشنهادی برای پاسخگویی: ۴ دقیقه

مسئله برش میله را در نظر بگیرید.

شبه‌کد زیر را به دقت بخوانید و آنرا طوری کامل کنید تا به درستی، یک الگوریتم برنامه‌ریزی پویا برای حل این مسئله به دست آید:

```
Cut-Rod ( p , n )
let r[ 0..n ] be a new array
r[ 0 ] = 0
for j = 1 to n
  r[ j ] = 
  for i = 1 to (j-1)
    r[ j ] = max ( r[ j ],  )
return 
```

پاسخ 1:

پاسخ دادید

0

پاسخ درست

 $p[j]$

پاسخ 2:

پاسخ درست

 $r[i] + r[j-i]$

پاسخ دادید

 $p[i] + r[j-i]$

پاسخ 3:

ادرست است

 $r[n]$

مره آزمون: 6.17 از 13