افرازهای عدد صحیح

تعریف، تولید و افرازهای مقید

امیرحسین محمودی



$$7 = 4 + 2 + 1$$

افراز عدد صحیح چیست؟

افرازِ یک عدد صحیح مثبت، نوشتن آن به‌صورت مجموع اجزای صحیح مثبت، بدون توجه به ترتیب است.

$$m = a_1 + a_2 + \cdots + a_N$$
$$a_i \geq 1$$

نمادها

نماد m: مجموعِ موردنظر

نماد N: تعداد اجزا

نماد a_i: اندازهٔ جزء شمارهٔ i

اجزا را می‌توان با هر ترتیبی نوشت.

مثال: یک افراز از ۷

$$a = [4, 2, 1], N = 3$$
$$7 = 4 + 2 + 1$$

هر جزء را مانند ارزش یک سکه در نظر بگیرید.

ترتیب مهم نیست؛ اجزای برابر و افراز تک‌جزئی نیز مجازند.

افرازهای ۶: یازده پاسخ متمایز

افراز تک‌جزئیِ ۶ نیز یک پاسخ است؛ سکه‌ها اجزا را نشان می‌دهند.



$6 = 5 + 1$



$6 = 4 + 1 + 1$



$6 = 3 + 2 + 1$



$6 = 2 + 2 + 2$



$6 = 2 + 1 + 1 + 1 + 1$

$6 = 6$

$6 = 4 + 2$

$6 = 3 + 3$

$6 = 3 + 1 + 1 + 1$

$6 = 2 + 2 + 1 + 1$

$6 = 1 + 1 + 1 + 1 + 1 + 1$

$$P(6) = 11$$

افراز و ترکیب مرتب: آیا ترتیب اجزا مهم است؟

افراز: فقط اندازهٔ اجزا مهم است



$$4+2 \equiv 2+4$$

یک افراز؛ جابه‌جایی اجزا یا سکه‌ها پاسخ تازه نمی‌سازد.

ترکیب مرتب: ترتیب هم مهم است



$$(4,2) \neq (2,4)$$

اگر ترتیب اجزا را ثبت کنیم، دو ترکیب مرتب متفاوت داریم.

$$m = a_1 + a_2 + \cdots + a_N, a_i \geq 1$$

فرم استاندارد: اجزای غیرصعودی

فرم استاندارد یعنی نوشتن اجزای افراز به‌ترتیب غیرصعودی؛ از بزرگ‌ترین به کوچک‌ترین، با امکان تکرار.

تعریف فرم استاندارد

$$a_1 \geq a_2 \geq \cdots \geq a_N \geq 1$$

هر جزء، بزرگ‌تر یا مساوی جزء بعدی است.

اجزای برابر مجازند؛ ترتیب، اکیداً نزولی نیست.

همان افراز، پیش و پس از مرتب‌سازی



مرتب‌سازی غیرصعودی



$$7 = 3 + 2 + 1 + 1$$

فقط ترتیب نوشتن عوض شده است، نه خود افراز. مولد هنگام ساخت پاسخ‌ها همین فرم را حفظ می‌کند.

شمارش افرازها با تولید آن‌ها فرق دارد

تابع P(m)، تعداد افرازهای عدد m است.

$$P\left(m\right) \sim \frac{e^{\pi\sqrt{\frac{2m}{3}}}}{4m\sqrt{3}}$$

رشد از هر چندجمله‌ای ثابت سریع‌تر است؛ اما نسبت به مجموع، زیرنمایی است.

نماد ∽ یعنی نسبت دو طرف با بزرگ‌شدن m به ۱ میل می‌کند؛ این برابری دقیق نیست.

$$P\left(1\right), P\left(2\right), P\left(3\right), P\left(4\right), P\left(5\right), P\left(6\right) = 1, 2, 3, 5, 7, 11$$

برای P(m) فرمول بستهٔ ساده‌ای با توابع ابتدایی شناخته نشده است؛ اما رشد مجانبی آن معلوم است و در بالا آمده است. محاسبهٔ دقیق تعداد همچنان ممکن است.

۳.۱ · تولید همهٔ افرازها: هدف و ایده

هدف: برای عدد صحیح مثبت m، همهٔ افرازهای آن را در فرم استاندارد و هرکدام دقیقاً یک بار تولید کنیم.

چگونه آن‌ها را تولید می‌کنیم؟

۱. هر مقدار مجاز برای جزء بعدی را امتحان می‌کنیم.

۲. مجموع باقی‌مانده را به‌صورت بازگشتی افراز می‌کنیم.

۳. جزءهای بعدی را از جزء فعلی بزرگ‌تر نمی‌گیریم تا فرم استاندارد حفظ شود.

وقتی باقی‌مانده صفر شد، افراز کامل را خروجی می‌دهیم.

حالت لازم برای اجرای این ایده

$$\mathrm{Rec}\,(\,r, B, N\,)$$

راهنمای متغیرهای افراز

`m` عدد هدف، برابر مبلغ مثال	`r` مجموع باقی‌مانده
`B` کران جزء بعدی	`N` تعداد اجزای انتخاب‌شده
`a[1..N]` اجزای انتخاب‌شده، یعنی ارزش سکه‌ها	`i` جزء بعدیِ در حال بررسی

ناوردا: مجموع اجزای فعلی به‌اضافهٔ r برابر m است؛ آغاز: r=m, B=m, N=0.

۳.۱ · تولید بازگشتی همهٔ افرازها

معنای فراخوانی Rec(r, B, N): همهٔ راه‌های کامل‌کردن پیشوند فعلی a[1..N] را تولید کن.

هر جزء مثبت بعدی را که هم با کران و هم با باقی‌مانده سازگار است، امتحان می‌کنیم:

$$1 \le i \le \min(B, r)$$

مقدار a[N+1] = i را می‌گذاریم؛ باقی‌مانده کم می‌شود، کران بعدی i می‌شود و تعداد اجزا یک واحد زیاد می‌شود.

$$\mathrm{Rec}(r - i, i, N + 1)$$

حالت پایه: اگر r = 0 شد، فقط a[1..N] را خروجی می‌دهیم و برمی‌گردیم.

$$\mathrm{Rec}(m, m, 0)$$

آغاز: پیشوند خالی و کل مجموع باقی است. **ناوردا:** مجموع اجزای فعلی به‌اضافهٔ r برابر m است.

```
procedure Rec(r, B, N):
    if r == 0:
        output a[1..N]
        return
    for i = 1 to min(B, r):
        a[N + 1] = i
        Rec(r - i, i, N + 1)

Rec(m, m, 0)
```

راهنمای متغیرهای افراز

- `m` عدد هدف، برابر مبلغ مثال
- `r` مجموع باقی‌مانده
- `B` کران جزء بعدی
- `N` تعداد اجزای انتخاب‌شده
- `a[1..N]` اجزای انتخاب‌شده، یعنی ارزش سکه‌ها
- `i` جزء بعدیِ در حال بررسی

ردگیری تولید افرازهای ۴: پنج خروجی

از جزء ۱ شروع می‌کنیم و پس از پایان هر شاخه برمی‌گردیم؛ سکه‌ها اندازهٔ اجزا را نشان می‌دهند.



$$4 = 1 + 1 + 1 + 1$$



$$4 = 2 + 1 + 1$$



$$4 = 2 + 2$$



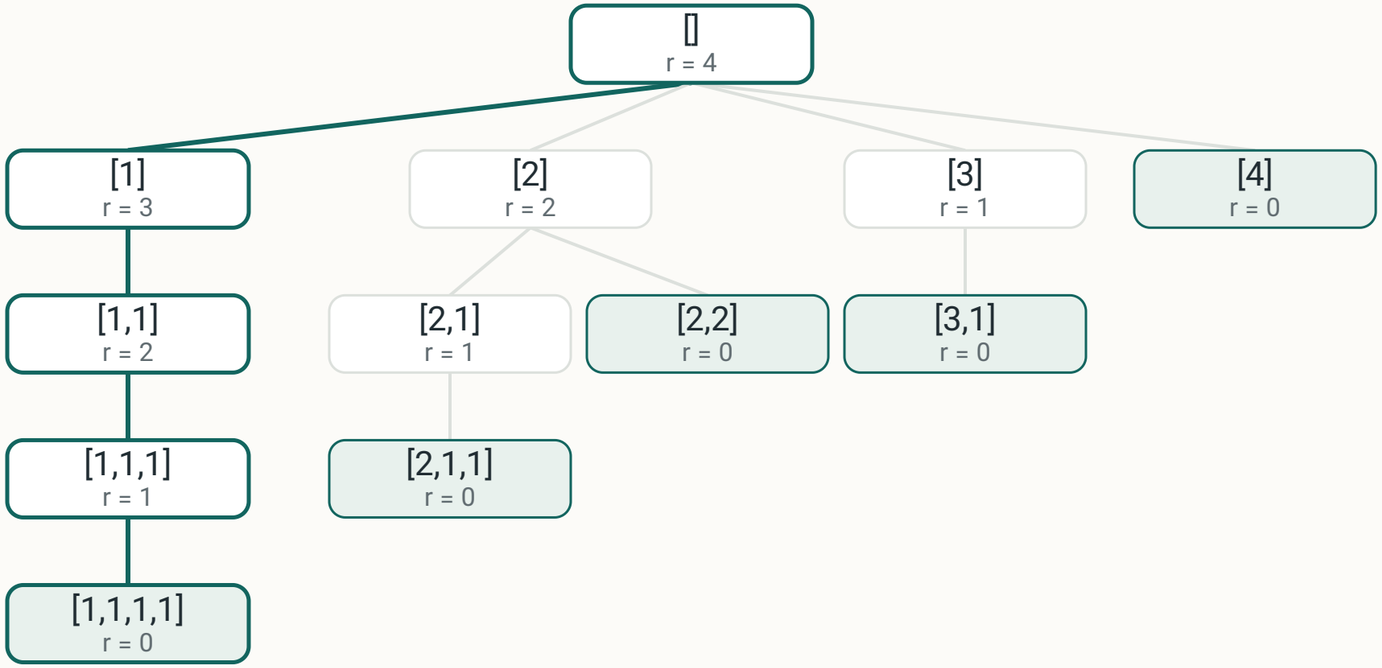
$$4 = 3 + 1$$



$$4 = 4$$

۳.۱ · از درخت بازگشت تا کران زمان

مثال: همهٔ فراخوانی‌های Rec(4,4,0). هر گره، پیشوند انتخاب‌شده و باقی‌مانده را نشان می‌دهد؛ برگ‌های سبز، خروجی‌اند.



برگ‌ها و طول مسیرها را بشمار

نماد V تعداد گره‌ها یا فراخوانی‌هاست؛ اینجا V=12 و تعداد برگ‌ها P(4)=5 است.

مسیر سبز ۴ انتخاب دارد. در حالت کلی، طول مسیر حداکثر m است، چون هر جزء دست‌کم ۱ است.

هر گره روی یک مسیرِ ریشه تا برگ قرار دارد. جمع طول همهٔ مسیرها، تعداد گره‌ها را از بالا کران می‌زند.

$$V \le 1 + mP(m)$$

$$T(m) = O(mP(m))$$

جست‌وجو O(V) کار دارد؛ نوشتن خروجی‌ها نیز حداکثر mP(m) جزء است. این‌ها کران بالا هستند، نه تعداد دقیق عملیات.

افرازهای عدد صحیح · تعریف و تولید

درستی مولد افراز: اعتبار، کامل‌بودن و یکتایی

هر خروجی معتبر است

اجزا مثبت‌اند؛ مجموعشان عدد ورودی است و فرم استاندارد دارند.

همهٔ پاسخ‌ها ساخته می‌شوند

جزء بعدیِ هر افراز معتبر، در بازهٔ انتخاب الگوریتم جا دارد.

هیچ پاسخ تکراری نداریم

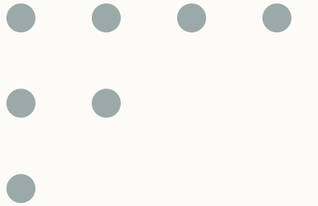
هر افراز استاندارد، فقط یک مسیر انتخاب دارد.

پایان‌پذیری: با هر جزء مثبت، باقی‌مانده دست‌کم یک واحد کم می‌شود.

نمودار فررز–یانگ و افراز مزدوج

افراز اولیه (Partition)

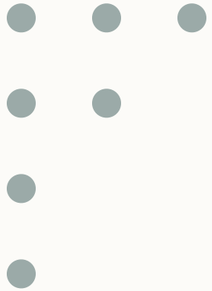
$a = [4, 2, 1]$





افراز مزدوج (Conjugate partition)

$a^* = [3, 2, 1, 1]$





هر جزء = ارزش یک سکه؛ هر افراز = یک فهرست سکه. نام این نمایش: Ferrers–Young diagram.

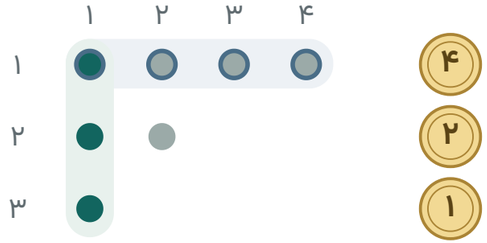
هر سطر به‌اندازهٔ یک جزء نقطه دارد؛ هر دو نمودار هفت نقطه دارند.

چرا ستون اول، تعداد اجزا را می‌شمارد؟

هر جزء مثبت است و سطرها از چپ شروع می‌شوند؛ پس هر سطر یک نقطه در ستون اول دارد.

ورودی: ۳ جزء، یعنی ۳ سطر

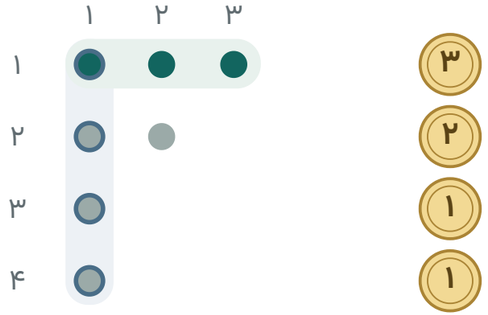
$$a = [\,4, 2, 1\,]$$



هر سکه یک جزء است؛ سه سکه، سه سطر می‌سازند.

مزدوج: بزرگ‌ترین جزء برابر ۳

$$b = [\,3, 2, 1, 1\,]$$



سه نقطهٔ ستون اول، سطر اولِ مزدوج را می‌سازند.

قضیهٔ ۳.۱: تعداد افرازهای m با دقیقاً n جزء، برابر تعداد افرازهای m با بزرگ‌ترین جزء n است.

دلیل: مزدوج‌گیری یک تناظر یک‌به‌یک و وارون‌پذیر است؛ با دو بار جابه‌جایی سطر و ستون به افراز اول برمی‌گردیم.

نماد b_j یعنی چه؟ یک ستون را بشماریم

$$b_j = \left| \left\{ i : a_i \geq j \right\} \right|$$

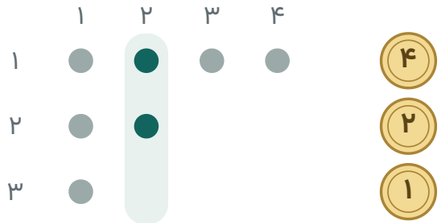
جزء جدید j = تعداد سطرهایی که تا ستون j می‌رسند.

راهنمای همهٔ نمادها

a_i	اندازهٔ جزء i، یعنی ارزش سکه؛ a_1 بزرگ‌ترین است.
i	شمارهٔ سطر ورودی
N	تعداد کل اجزای ورودی، یعنی سطرها
j	شمارهٔ ستونی که می‌شماریم
b_j	جزء مزدوج j: ارتفاع ستون و ارزش سکهٔ تازه
$\lvert S \rvert$	تعداد اعضای مجموعهٔ S
$\text{len}(b)$	تعداد اجزای افراز مزدوج

مثال: ستون دوم را بشمار

$$a = [4, 2, 1], j = 2$$



$$b_2 = \left| \{1, 2\} \right| = 2$$

سطرهای ۱ و ۲ می‌رسند؛ سطر ۳ نمی‌رسد.

$b_1 = N = 3$ هر سطر در ستون اول نقطه دارد.

$\text{len}(b) = a_1 = 4$ هر ستون، یک جزء تازه می‌سازد.

افرازهای عدد صحیح · تعریف و تولید

۳.۲ · افراز مزدوج از ارتفاع ستون‌ها

هدف: تبدیل یک افراز به مزدوج آن، با حفظ مجموع.

$$a = [4, 2, 1]$$



افراز مزدوج



جزء نخست، ارتفاع ستون‌ها را با ۱ آغاز می‌کند؛ هر جزء بعدی، ارتفاع ستون‌های زیر خود را یک واحد می‌افزاید.

```
procedure ConjPartition(a[1..N]):
    for j = 1 to a[1]:
        b[j] = 1
    nPrime = a[1]
    for i = 2 to N:
        for j = 1 to a[i]:
            b[j] = b[j] + 1
    return b[1..nPrime]
```

راهنمای متغیرهای افراز

`a[i]` جزء ورودی i، یعنی ارزش سکه	`N` تعداد اجزای ورودی
`b[j]` جزء مزدوج j، برابر ارتفاع ستون	`i` شمارهٔ سطر ورودی
`j` شمارهٔ ستون	`nPrime` تعداد اجزای خروجی، برابر a[1]

افرازهای ۴: بزرگ‌ترین جزء یا تعداد اجزا؟

بزرگ‌ترین جزء دقیقاً ۲

$[2,2]$	۲ جزء
$[2,1,1]$	۳ جزء

هر جزء حداکثر ۲ است و دست‌کم یک جزء برابر ۲ داریم؛ پس افراز [1,1,1,1] پذیرفته نیست.

تعداد اجزا دقیقاً ۲

$[2,2]$	بیشینهٔ ۲
$[3,1]$	بیشینهٔ ۳

دو جزء می‌خواهیم؛ لازم نیست بزرگ‌ترین جزء برابر ۲ باشد.

$$[2,1,1] \leftrightarrow [3,1]$$

مزدوج، خانوادهٔ قیدِ بیشینه را به خانوادهٔ قیدِ تعداد می‌برد؛ پس می‌توان مولد را دوباره استفاده کرد.

افرازهای عدد صحیح · تعریف و تولید

شمارش بر اساس تعداد اجزا

نماد P(m,n)، تعداد افرازهای m با دقیقاً n جزء است؛ در مثال، n سکه داریم.

$$P(m) = \sum_{n=1}^{m} P(m,n)$$

هر افراز، تعداد مشخصی جزء دارد؛ این گروه‌ها هم‌پوشانی ندارند و همهٔ افرازها را پوشش می‌دهند.

$$P(6) = 1 + 3 + 3 + 2 + 1 + 1 = 11$$

به‌ترتیب برای تعداد اجزای ۱ تا ۶.

حالت‌های پایه؛ برای m مثبت

$$P(m,1) = P(m,m) = 1$$

$$P(m,0) = 0, P(0,0) = 1$$

دو و سه جزء: فرمول‌های دقیق

دقیقاً دو جزء

$$P(m,2)=\left\lfloor \frac{m}{2} \right\rfloor$$

نمونه: P(6,2) = 3؛ پاسخ‌ها: [5,1]، [4,2] و [3,3].

دقیقاً سه جزء

$$P(m,3)=\left\lfloor \frac{m^2+4}{12} \right\rfloor$$

نمونه: P(6,3) = 3؛ پاسخ‌ها: [4,1,1]، [3,2,1] و [2,2,2].

نماد ⌊x⌋ یعنی بزرگ‌ترین عدد صحیحِ کوچک‌تر یا مساوی x.

فقط برای n ثابت، وقتی m بزرگ می‌شود:

$$P(m,n)=\Theta\left(\, m^{n-1} \,\right)$$

مزدوج‌گیری همین تعداد را برای بزرگ‌ترین جزء n می‌دهد؛ در مثال، بزرگ‌ترین ارزش سکه n است.

شمارش دقیق: یک مثال، دو گروه

مثال: افرازهای ۶ با دقیقاً ۳ جزء را می‌شماریم؛ هر افراز فقط در یکی از دو گروه زیر است.

$$P(m,n) = P(m-1,n-1) + P(m-n,n)$$

گروه اول: دست‌کم یک جزء ۱

فقط یک جزء ۱ را حذف کن: مجموع یک واحد و تعداد اجزا یک واحد کم می‌شود.

$$[4,1,1] \to [4,1]$$
$$[3,2,1] \to [3,2]$$
$$P(5,2) = 2$$

گروه دوم: همهٔ اجزا دست‌کم ۲

از هر سه جزء، یک واحد کم کن: مجموع ۳ واحد کم می‌شود؛ تعداد اجزا همان ۳ می‌ماند.

$$[2,2,2] \to [1,1,1]$$
$$P(3,3) = 1$$

$$P(6,3) = P(5,2) + P(3,3) = 2 + 1 = 3$$

یک افراز تهی داریم: فهرست خالی، بدون جزء و با مجموع صفر. عدد ۱، تعداد آرایش‌هاست، نه تعداد اجزا.

$$P(0,0) = 1$$

با حذف ۱ از افراز [1] به فهرست خالی می‌رسیم؛ این یک پاسخ معتبر است. حالت‌های ناممکن، تعداد صفر دارند.

$$P(1,1) = P(0,0) + P(0,1) = 1 + 0 = 1$$

۳٫۱ و ۳٫۳: فقط مقداردهی اولیه عوض می‌شود

۳٫۱ — شروع آزاد

همهٔ افرازهای عدد m
پیش‌شرط: $m \geq 1$

```
GenerateAll(m):
    Rec(m,m,0)
```

نمونهٔ m=7, n=3: (r,B,N) = (7,7,0)

۳٫۳ — شروع تغییریافته

بزرگ‌ترین جزء (ارزش سکه) دقیقاً n
پیش‌شرط: $1 \leq n \leq m$

```
GenerateLargest(m,n):
    a[1] = n
    Rec(m-n,n,1)
```

نمونهٔ m=7, n=3: (r,B,N) = (4,3,1)؛ نخست a[1] = 3

روال مشترک — بدون تغییر

```
Rec(r,B,N):
    if r == 0:
        output a[1..N]
    else:
        for i = 1..min(B,r):
            a[N+1] = i
            Rec(r-i,i,N+1)
```

راهنمای نمادها

- **m** مجموع هدف، برابر مبلغ مثال
- **r** مجموع باقی‌مانده
- **N** تعداد اجزای انتخاب‌شده
- **i** جزء بعدیِ پیشنهادی
- **n** بزرگ‌ترین جزء، یعنی ارزش سکه
- **B** سقف جزء بعدی
- **a** آرایهٔ اجزا (ارزش سکه‌ها)

دقیقاً، نه حداکثر: Rec(m,n,0) کافی نیست؛ ممکن است هیچ جزء n انتخاب نکند.

۳.۳ · افراز با بزرگ‌ترین جزء ۳؛ سکهٔ ۳ واحدی

هدف: تولید افرازهای ۷ با بزرگ‌ترین جزء دقیقاً ۳.

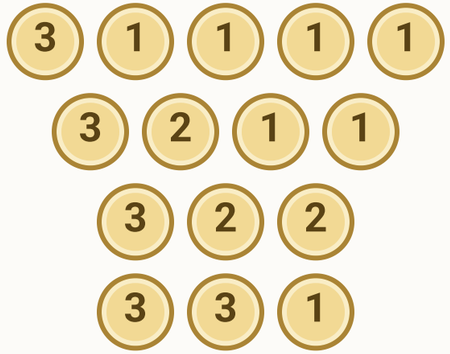
با انتخاب a[1] = n، وجود جزء n تضمین می‌شود. کران n برای بقیهٔ اجزا، بیشینه را دقیقاً n می‌کند.

```
a[1] = n
Rec(m - n, n, 1)

a[1] = 3
Rec(4, 3, 1)
```

راهنمای متغیرهای افراز

`m` عدد هدف	`n` بزرگ‌ترین جزء موردنظر
`a[1] = n` ابتدا جزء n را انتخاب می‌کنیم	`r = m − n` مجموع باقی‌مانده
`B = n` جزء بعدی از n بزرگ‌تر نیست	`N = 1` یک جزء انتخاب شده است



چهار افراز خروجی؛ تعداد اجزا لازم نیست برابر باشد.

افرازهای عدد صحیح · تعریف و تولید

۳.۴ · تولید افراز با دقیقاً ۳ جزء

هدف: افرازهای m با دقیقاً n جزء؛ مثال: m=7, n=3.

```
procedure RecPartition2(r, B, N):
    if r == 0:
        b = ConjPartition(a[1..N])
        output b
    else:
        for i = 1 to min(B, r):
            a[N+1] = i
            RecPartition2(r-i, i, N+1)

a[1] = n
RecPartition2(m-n, n, 1)
```

راهنمای متغیرهای افراز

- `m` مجموع اولیهٔ موردنظر
- `n` تعداد اجزای خروجی؛ $1 \le m \le n$
- `r` مجموع باقی‌ماندهٔ افراز میانی
- `B` کران جزء بعدی
- `N` تعداد اجزای فعلیِ افراز میانی
- `a[1..N]` پیشوند میانی؛ نخستین جزء آن n است
- `i` جزء بعدیِ در حال بررسی
- `b` افراز مزدوجِ خروجی با دقیقاً n جزء

برای $1 \le n \le m$: آغاز مانند ۳.۳ است؛ در برگ، به‌جای a، مزدوج آن را خروجی می‌دهیم.

تبدیل قید: از بزرگ‌ترین جزء ۳ به دقیقاً ۳ جزء

ورودی‌های میانی: بزرگ‌ترین جزء ۳؛ خروجی‌های مزدوج: دقیقاً ۳ جزء.



$$[3,1,1,1,1] \to [5,1,1]$$



$$[3,2,1,1] \to [4,2,1]$$



$$[3,2,2] \to [3,3,1]$$



$$[3,3,1] \to [3,2,2]$$

افرازهای عدد صحیح · تعریف و تولید

چرا این پیچیدگی‌ها؟ الگوریتم‌های ۳.۱ تا ۳.۴

نماد P(m)، تعداد همهٔ افرازهاست؛ P(m,n)، تعداد افرازهای با دقیقاً n جزء است.

الگوریتم و کار آن	چرا این کران؟	زمان	حافظه
۳.۱ · همهٔ افرازها	هر مسیر حداکثر m جزء دارد؛ تعداد برگ‌ها P(m) است. مجموع طول مسیرها، کار جست‌وجو و نوشتن خروجی‌ها را کران می‌زند.	$O(mP(m))$	$O(m)$
۳.۲ · یک مزدوج	ابتدا a_1 خانه مقداردهی و سپس $m-a_1$ افزایش انجام می‌شود: در مجموع m عمل. آرایهٔ خروجی a_1 خانه دارد.	$\Theta(m)$	$\Theta(a_1)$
۳.۳ · بیشینهٔ n	همان جست‌وجوی ۳.۱، ولی فقط با P(m,n) برگ؛ برابری تعدادها از مزدوج‌گیری به دست می‌آید.	$O(mP(m,n))$	$O(m)$
۳.۴ · دقیقاً n جزء	جست‌وجوی ۳.۳، به‌اضافهٔ یک مزدوج‌گیری Θ(m) برای هر یک از P(m,n) خروجی‌ها.	$O(mP(m,n))$	$O(m)$

فرض: عملیات حسابی ثابت‌زمان و خروجی‌ها یکی‌یکی مصرف می‌شوند. حافظه شامل پشته و خروجی جاری است، نه ذخیرهٔ همهٔ پاسخ‌ها. کران‌های O، کران بالا هستند.

افرازهای عدد صحیح · تعریف و تولید

از هدف تا پاسخ: ترکیب الگوریتم‌ها

هدف: همهٔ افرازهای ۷ با دقیقاً ۳ جزء را تولید کنیم.

گام اول · ۳.۳ با بازگشتِ ۳.۱

جزء ۳ را از قبل انتخاب کن؛ چهار واحد باقی‌مانده را با کران ۳ افراز کن.

$$a_1 = 3, \mathrm{Rec}\left(4, 3, 1 \right)$$

گام دوم · ۳.۲ روی هر خروجی

هر افراز میانی با بزرگ‌ترین جزء ۳ را مزدوج کن؛ خروجی دقیقاً ۳ جزء دارد.

$$b = \mathrm{ConjPartition}\left(a \right)$$

الگوریتم ۳.۴ همین ترکیب است: در هر برگِ جست‌وجوی ۳.۳، به‌جای a، مزدوج آن را خروجی بده.

$$[3,1,1,1,1] \to [5,1,1]$$

$$[3,2,2] \to [3,3,1]$$

$$[3,2,1,1] \to [4,2,1]$$

$$[3,3,1] \to [3,2,2]$$

سمت چپ: تمام افرازهای بیشینهٔ ۳؛ سمت راست: تمام افرازهای دقیقاً ۳ جزئی. تناظر یک‌به‌یک، کامل‌بودن و نبود تکرار را حفظ می‌کند.

Kreher & Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, 1999.