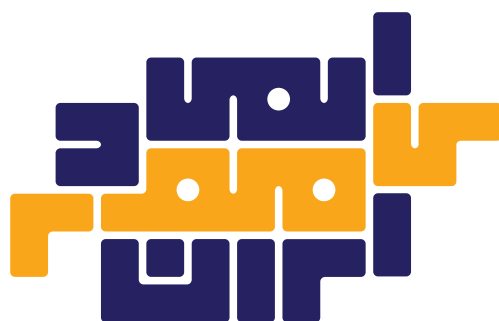




## المپیاد کامپیوتر ایران

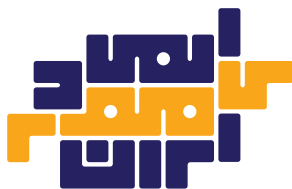
---

### آزمون نهایی دوره ی تابستان سی و ششمین المپیاد کامپیوتر ایران

آزمون عملی سوم - ۳۰ شهریور ۱۴۰۵

---

- آزمون شامل سه سوال است. زمان آزمون ۵ ساعت است.
- برای هر سوال می توانید حداکثر ۵۰ ارسال انجام دهید.
- بین هر دو ارسال متوالی برای یک سوال باید حداقل ۱ دقیقه فاصله باشد.
- موارد مبهم در سوالات را فقط از طریق قسمت Communication بپرسید.
- ترتیب سوالات به صورت کاملاً تصادفی انتخاب شده اند.
- نام کاربری برای ورود به سیستم day3 و رمز عبور goodluck است.



## خانندان (clan)

در سرزمین اسپادانا،  $n$  خانندان با شماره‌های ۱ تا  $n$  زندگی می‌کنند. خانندان  $i$  ام دارای  $b_i$  پسر و  $g_i$  دختر است. این خانندان می‌خواهند با ازدواج میان فرزندان‌شان، یک اتحادیه‌ی واحد تشکیل دهند؛ یعنی در پایان فرایند، هر  $n$  خانندان عضو همان یک اتحادیه باشند. این فرایند طی  $n - 1$  سال انجام می‌شود.

در سال اول، یک پسر و یک دختر از دو خانندان متفاوت با یکدیگر ازدواج می‌کنند. این دو خانندان با همین ازدواج، اعضای اولیه‌ی همان اتحادیه‌ی واحد می‌شوند. از این پس، هر ازدواج فقط می‌تواند باعث پیوستن یک خانندان جدید به همین اتحادیه شود و اتحادیه‌ی دیگری تشکیل نمی‌شود.

در هر یک از سال‌های ۲ تا  $n - 1$ ، دقیقاً یک خانندان که هنوز عضو اتحادیه نیست انتخاب می‌شود. سپس یکی از فرزندان این خانندان با فردی از جنس مخالف از میان فرزندان خانندان‌هایی که پیش‌تر عضو اتحادیه شده‌اند ازدواج می‌کند. با این ازدواج، خانندان جدید نیز به همان اتحادیه می‌پیوندند. در طول کل فرایند، هر فرد می‌تواند حداکثر در یک ازدواج شرکت کند.

بزرگ خانندان  $i$  ام شرط گذاشته است که خانندان‌ش باید حداکثر تا پایان سال  $i$  به اتحادیه بپیوندند. به بیان دیگر، اگر خانندان  $i$  ام در سال  $t$  عضو اتحادیه شود، باید داشته باشیم:

$$t \leq i.$$

برای هر سناریو، بسته به نوع پاسخ مورد انتظار، یکی از دو مسئله‌ی زیر را حل کنید:

- آیا می‌توان با رعایت شرط تمام بزرگ خانندان‌ها، ترتیب ازدواج‌ها را به‌گونه‌ای انتخاب کرد که هر  $n$  خانندان در نهایت عضو اتحادیه شوند؟
- اگر چنین چیزی ممکن باشد، چند ترتیب متفاوت برای پیوستن خانندان‌ها به اتحادیه وجود دارد؟

دو روش را متفاوت در نظر می‌گیریم اگر در حداقل یکی از سال‌های ۲ تا  $n - 1$ ، خاندانی که در آن سال به اتحادیه می‌پیوندد در دو روش متفاوت باشد. بنابراین، انتخاب افراد مشخصی که با یکدیگر ازدواج می‌کنند در تفاوت دو روش نقشی ندارد؛ تنها ترتیب پیوستن خانندان‌ها مهم است.

## ورودی

در خط اول عدد صحیح  $T$ ، تعداد سناریوهای مستقل، آمده است.

در خط اول هر سناریو، دو عدد صحیح  $n$  و  $c$  آمده است:  $n$  تعداد خاندان‌ها و  $c$  نوع پاسخ مورد انتظار برای آن سناریو را مشخص می‌کند.

در هر یک از  $n$  خط بعدی، دو عدد صحیح  $b_i$  و  $g_i$  آمده است که به ترتیب تعداد پسرها و دخترهای خاندان  $i$ ام را نشان می‌دهند.

$T$   
 $n \ c$   
 $b_1 \ g_1$   
 $\dots$   
 $b_n \ g_n$

## خروجی

به ازای هر سناریو، بسته به مقدار  $c$ ، یکی از موارد زیر را در یک خط چاپ کنید:

- اگر  $c = 0$  باشد، در صورتی که تشکیل اتحادیه با رعایت تمام شرط‌ها ممکن باشد، عبارت YES و در غیر این صورت عبارت NO را چاپ کنید.
- اگر  $c = 1$  باشد، تعداد روش‌های متفاوت برای تشکیل اتحادیه را به پیمانه‌ی ۹۹۸۲۴۴۳۵۳ چاپ کنید.

## محدودیت‌ها

- $1 \leq T \leq 3 \times 10^5$
- $2 \leq n \leq 3 \times 10^5$
- $c \in \{0, 1\}$
- $0 \leq b_i, g_i \leq 10^9$
- $b_i + g_i > 0$
- مجموع مقادیر  $n$  به ازای تمام سناریوها حداکثر  $3 \times 10^5$  است.

## زیرمسئله‌ها

| محدودیت‌ها                    | نمره | زیرمسئله |
|-------------------------------|------|----------|
| ورودی و خروجی نمونه           | ۰    | ۱        |
| $\sum n \leq 1400, n \leq 14$ | ۸    | ۲        |
| $g_i \geq 1, b_i \geq 1$      | ۶    | ۳        |
| $\sum n \leq 5000, c = 0$     | ۱۳   | ۴        |
| $c = 0$                       | ۱۲   | ۵        |
| $\sum n \leq 5000$            | ۱۲   | ۶        |
| $b_i + g_i \geq 2$            | ۱۷   | ۷        |
| بدون محدودیت اضافی            | ۳۲   | ۸        |

## ورودی و خروجی نمونه

| ورودی استاندارد | خروجی استاندارد |
|-----------------|-----------------|
| 3               | YES             |
| 3 0             | 1               |
| 0 1             | NO              |
| 1 0             |                 |
| 1 1             |                 |
| 3 1             |                 |
| 0 1             |                 |
| 1 0             |                 |
| 1 1             |                 |
| 4 0             |                 |
| 0 1             |                 |
| 0 1             |                 |
| 1 1             |                 |
| 2 0             |                 |

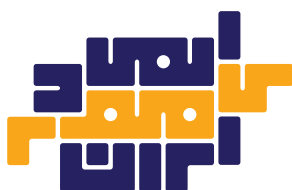
## شرح ورودی و خروجی نمونه

سناریوهای اول و سوم دارای  $c = 0$  هستند و سناریوی دوم دارای  $c = 1$  است.

در سناریوی اول، در سال اول پسر خاندان ۳ با دختر خاندان ۱ ازدواج می‌کند. در نتیجه، خاندان‌های ۳ و ۱ اعضای اولیه‌ی اتحادیه‌ی واحد می‌شوند. سپس در سال دوم، پسر خاندان ۲ با دختر دیگری از خاندان ۳ ازدواج می‌کند و خاندان ۲ نیز به همان اتحادیه می‌پیوندد. بنابراین سال پیوستن خاندان‌های ۱، ۲ و ۳ به ترتیب ۱، ۲ و ۱ است که همگی شرط مربوط به شماره‌ی خود را رعایت می‌کنند.

سناریوی دوم دقیقاً همان خاندان‌های سناریوی اول را دارد، با این تفاوت که  $c = 1$  است. در این حالت تنها یک ترتیب متفاوت برای پیوستن خاندان‌ها وجود دارد: ابتدا خاندان‌های ۳ و ۱ و سپس خاندان ۲ به اتحادیه می‌پیوندد؛ بنابراین پاسخ برابر با ۱ است.

می‌توان نشان داد که در سناریوی سوم، هیچ ترتیب معتبری برای پیوستن همه‌ی خاندان‌ها به اتحادیه وجود ندارد.



## شونه به شونه (comb)

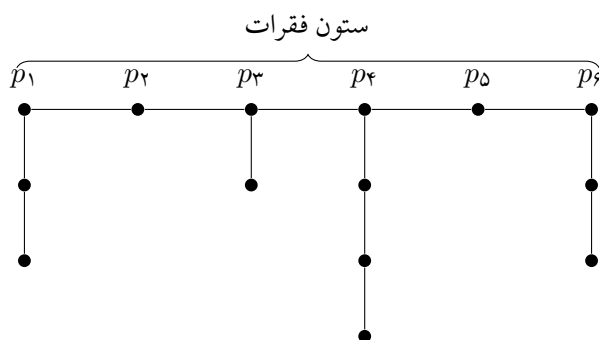
«کوچه به کوچه، خونه به خونه، میام محله‌ها تون!»

— مسعود خاموش‌پژوه

آقای خاموش‌پژوه، مجری مسابقه‌ی تلویزیونی معروف، که از دیدار پیشین خود با دانش‌پژوهان تجربه و خاطره‌ی بسیار خوبی داشته، تصمیم گرفته است تا دوباره با دانش‌پژوهان تجدید دیدار کند و از آن‌ها بخواهد بار دیگر برای او شیرین‌کاری کنند. او برای شیرین‌کاری امسال، این بازی را در نظر گرفته است: شما باید تمام برگ‌های یک درخت شانه‌ای در ذهن او را با پرسیدن تعدادی پرسش عجیب حدس بزنید!

به یک درخت  $T$ ، **درخت شانه‌ای** می‌گوییم اگر مسیری به شکل  $P = p_1 p_2 \dots p_k$  در آن وجود داشته باشد که  $T$  از  $P$  به روش زیر ساخته شده باشد:

برای هر رأس  $p_i$ ، می‌توان حداکثر یک مسیر به شکل  $Q_i = q_{i,1} q_{i,2} \dots q_{i,\ell_i}$  به آن آویزان کرد؛ به این صورت که یال  $p_i q_{i,1}$  به درخت اضافه شود ( $\ell_i \geq 1$ ). مسیرهای  $Q_i$  دوه‌دو اشتراک رأسی ندارند و هیچ رأسی از آن‌ها روی  $P$  قرار ندارد. ممکن است برای بعضی از رئوس  $p_i$  هیچ مسیری آویزان نشده باشد ( $\ell_i = 0$ ). مسیر  $P$  را **ستون فقرات** درخت می‌نامیم. به عنوان مثال، شکل زیر یک درخت شانه‌ای را نشان می‌دهد.



آقای خاموش‌پژوه در ذهن خود یک درخت شانه‌ای در نظر گرفته است. شما در هر مرحله می‌توانید چهار رأس  $a, b, c, d$  را که لزوماً متمایز نیستند انتخاب کنید و پرسش زیر را از او بپرسید: مسیر یکتای بین  $a$  و  $b$  را با  $P(a, b)$  و مسیر یکتای بین  $c$  و  $d$  را با  $P(c, d)$  نشان می‌دهیم. آیا این دو مسیر رأس مشترکی دارند؟ اگر پاسخ منفی باشد، آقای خاموش‌پژوه اعلام می‌کند که دو مسیر با یکدیگر برخوردی ندارند. در غیر این صورت، او رأسی مانند  $v$  را برمی‌گرداند که اولین رأس مشترک دو مسیر هنگام پیمایش  $P(a, b)$  از  $a$  به سمت  $b$  است. به عبارت دیگر،  $v$  رأسی از اشتراک دو مسیر است که کمترین فاصله را از رأس  $a$  دارد.

هدف شما پیدا کردن اندیس تمام برگ‌های درخت شانه‌ای آقای خاموش‌پژوه است (رئوسی با درجه‌ی ۱). هرچه بتوانید این کار را با تعداد پرسش‌های کمتری انجام دهید، شیرین‌کاری شما در نظر آقای خاموش‌پژوه جذاب‌تر خواهد بود و امتیاز بهتری دریافت خواهید کرد!

دقت کنید که درختی که آقای خاموش‌پژوه در ذهن دارد، لزوماً در طول پرسش‌ها ثابت نمی‌ماند و او می‌تواند پس از هر پرسش آن را تغییر دهد؛ تنها تضمین این است که همواره حداقل یک درخت شانه‌ای وجود داشته باشد که با پاسخ تمام پرسش‌های داده‌شده تا آن لحظه سازگار باشد. همچنین، در هر مرحله می‌توانید از آقای خاموش‌پژوه بخواهید **ستون فقرات** درخت فعلی را به شما اعلام کند، اما در این صورت بخشی از امتیاز سؤال را از دست خواهید داد (به بخش نمره‌دهی توجه کنید).

## پیاده‌سازی

در این سؤال شما اجازه‌ی خواندن از ورودی یا نوشتن در خروجی را ندارید. تنها راه تبادل اطلاعات با استفاده از توابعی که در ادامه معرفی می‌کنیم خواهد بود. در صورت هرگونه تلاش برای خواندن از ورودی یا نوشتن در خروجی یا استفاده از تابع `exit` نمره‌ی شما صفر می‌شود.

```
std::vector<int> find_leaves(int n)
```

شما باید این تابع را پیاده کنید. در هر تست این تابع دقیقاً یک بار فراخوانی می‌شود و تعداد رأس‌های درخت، یعنی  $n$ ، به آن داده می‌شود. در این تابع می‌توانید با فراخوانی توابع `ask` و `get_backbone` درباره‌ی درخت اطلاعات کسب کنید. در پایان، باید تمام برگ‌های درخت را برگردانید. ترتیب برگ‌های برگردانده‌شده اهمیتی ندارد، اما هر برگ باید دقیقاً یک بار برگردانده شود.

•  $n$ : تعداد رأس‌های درخت.

```
int ask(int a, int b, int c, int d)
```

با فراخوانی این تابع می‌توانید دو مسیر  $P(a, b)$  و  $P(c, d)$  را بررسی کنید. اگر این دو مسیر هیچ رأس مشترکی نداشته باشند، مقدار ۱ – برگردانده می‌شود. در غیر این صورت، خروجی تابع اولین رأس مشترک این دو مسیر هنگام پیمایش مسیر  $P(a, b)$  از  $a$  به سمت  $b$  است؛ به عبارت دیگر، رأس مشترکی که کمترین فاصله را از  $a$  دارد. رئوس  $a, b, c$  و  $d$  لزوماً متمایز نیستند. محدودیتی برای تعداد فراخوانی‌های این تابع وجود ندارد، اما تعداد آن‌ها در نمره‌ی شما تأثیر دارد. اگر هر یک از آرگومان‌ها خارج از بازه‌ی ۱ تا  $n$  باشد، خطای `protocol violation` دریافت خواهید کرد.

•  $a$ : ابتدای مسیر اول.

•  $b$ : انتهای مسیر اول.

- c: ابتدای مسیر دوم.

- d: انتهای مسیر دوم.

```
std::vector<int> get_backbone()
```

- 

با فراخوانی این تابع، رئوس یک ستون فقرات معتبر درخت به ترتیب قرارگرفتن روی مسیر برگردانده می‌شوند. فراخوانی این تابع محدودیتی ندارد، اما حتی یک بار فراخوانی آن ضریب نمره‌ی مربوط به ستون فقرات را اعمال می‌کند. برای جزئیات بیشتر به بخش نمره‌دهی مراجعه کنید.

## ارزیاب نمونه

فایل‌های مربوط به سؤال را دانلود کنید. راه‌حل خود را درون comb.cpp پیاده‌سازی کنید. برای کامپایل و اجرای برنامه، compile\_cpp.sh را اجرا کنید. سپس فایل اجرایی comb را که توسط compile\_cpp.sh تولید شده است اجرا کنید.

## ورودی

در خط اول دو عدد  $n$  و  $k$ ، به‌ترتیب تعداد رأس‌ها و تعداد رأس‌های ستون فقرات، آمده است. در خط دوم، رأس‌های ستون فقرات به ترتیب روی مسیر نوشته شده‌اند. هر یک از  $n - 1$  خط بعدی شامل دو سر یک یال درخت است.

```
n k
p1 p2 ... pk
u1 v1
⋮
un-1 vn-1
```

## خروجی

در صورت اجرای موفق، ارزیاب نمونه در خط اول تعداد فراخوانی‌های ask و get\_backbone را چاپ می‌کند. در خط دوم تعداد برگ‌های برگردانده‌شده و در خط سوم شماره‌ی این برگ‌ها چاپ می‌شوند.

## محدودیت‌ها

- $2 \leq n \leq 1500$
- $1 \leq k \leq n$
- درخت ورودی یک درخت شانه‌ای با ستون فقرات اعلام شده است.

## زیرمسئله‌ها

| محدودیت‌ها                                        | نمره | زیرمسئله |
|---------------------------------------------------|------|----------|
| ورودی و خروجی نمونه                               | ۰    | ۱        |
| $n \leq 50$                                       | ۶    | ۲        |
| درخت یک مسیر است.                                 | ۶    | ۳        |
| حداکثر برای یکی از رئوس ستون فقرات $l_i > 0$ است. | ۸    | ۴        |
| برای هر $i$ ، $l_i \leq 1$ است.                   | ۱۸   | ۵        |
| بدون محدودیت اضافی                                | ۶۲   | ۶        |

## نمره‌دهی

فرایند زیر برای محاسبه‌ی نمره‌ی هر زیرمسئله به کار می‌رود. فرض کنید  $Q$  بیشترین تعداد پرسش تابع `ask` در یک تست از آن زیرمسئله باشد. ضریب تعداد پرسش‌ها برای نمره‌ی زیرمسئله برابر تابع زیر است:

$$score(Q) = \begin{cases} 1 & Q \leq 3010, \\ 0.95 - 0.15 \times \frac{Q - 3010}{750} & 3010 < Q \leq 3760, \\ 0.80 - 0.10 \times \frac{Q - 3760}{750} & 3760 < Q \leq 4510, \\ 0.70 - 0.10 \times \frac{Q - 4510}{3000} & 4510 < Q \leq 7510, \\ 0.25 & 7510 < Q \leq 2250100, \\ 0 & Q > 2250100. \end{cases}$$

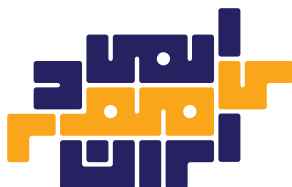
اگر نمره کامل زیرمسئله  $P$  باشد، در حالت عادی  $P \times \text{score}(Q)$  امتیاز می‌گیرید. اگر در هر تستی تابع `get_backbone` را فراخوانی کنید، تنها ۱۵ درصد امتیاز محاسبه شده برای همان تست را دریافت می‌کنید. بنابراین بیشترین امتیاز ممکن با استفاده از این تابع ۱۵ درصد است و این ضریب در نمره‌ی زیرمسئله نیز اعمال خواهد شد.

## ورودی و خروجی نمونه

| ورودی استاندارد                                                 | خروجی استاندارد   |
|-----------------------------------------------------------------|-------------------|
| 8 4<br>2 5 1 7<br>2 5<br>5 1<br>1 7<br>2 3<br>3 6<br>1 4<br>7 8 | 7 0<br>3<br>4 6 8 |

## شرح ورودی و خروجی نمونه

در نمونه، درخت و یک ستون فقرات معتبر آن به ارزیاب نمونه داده می‌شود. خروجی تنها برای بررسی محلی پروتکل است و ترتیب برگ‌های گزارش شده می‌تواند متفاوت باشد.



## کیک (cake)

یک قناد می‌خواهد یک کیک بزرگ بپزد و برای این کار به دقیقاً  $K$  کیلوگرم آرد نیاز دارد. در شهر  $n$  نانوائی وجود دارد که با  $n - 1$  خیابان به یکدیگر متصل شده‌اند و ساختاری درختی به ریشه‌ی نانوائی ۱ ساخته‌اند. نانوائی‌ها از ۱ تا  $n$  شماره‌گذاری شده‌اند و نانوائی  $i$  هر کیلوگرم آرد را به قیمت  $a_i$  تومان می‌فروشد.

در ابتدا انبار همه‌ی نانوائی‌ها خالی است. این ماجرا طی  $m$  روز دنبال می‌شود و در هر روز اتفاقات زیر می‌افتد:

۱. شارژ انبار: به انبار نانوائی  $v_j$  و تمام نانوائی‌های زیردرخت آن، هرکدام دقیقاً ۱ کیلوگرم آرد اضافه می‌شود.
۲. محاسبه‌ی هزینه: قناد می‌خواهد بداند با موجودی فعلی نانوائی‌ها، خرید  $K$  کیلوگرم آرد حداقل چقدر برایش خرج برمی‌دارد.

توجه داشته باشید که قناد در هر روز صرفاً قیمت را حساب می‌کند و هیچ خریدی انجام نمی‌دهد. بنابراین موجودی نانوائی‌ها در اثر این محاسبات کم نمی‌شود و آردها برای روزهای بعد در انبار باقی می‌مانند.

همچنین از هر نانوائی می‌توان به مقدار دلخواه (حداکثر تا سقف موجودی همان روز آن نانوائی) آرد تهیه کرد. اگر در یک روز مجموع آرد موجود در کل شهر کمتر از  $K$  کیلوگرم باشد، امکان تهیه‌ی آرد وجود ندارد و پاسخ ۱- است. به قناد کمک کنید تا به ازای هر یک از  $m$  روز، کمترین هزینه‌ی خرید  $K$  کیلوگرم آرد (یا ۱-) را به دست آورد.

## ورودی

سطر اول ورودی شامل سه عدد صحیح  $n$  و  $m$  و  $K$  است.

سطر دوم شامل  $n$  عدد صحیح  $a_1, a_2, \dots, a_n$  است که قیمت هر کیلوگرم آرد در نانوائی‌ها را نشان می‌دهد.

سطر سوم شامل  $n - 1$  عدد صحیح  $p_2, p_3, \dots, p_n$  است. عدد  $p_i$  پدر رأس  $i$  در درخت ریشه‌دار است.

سطر آخر شامل  $m$  عدد صحیح  $v_1, v_2, \dots, v_m$  است که نانوائی انتخاب‌شده در هر روز را مشخص می‌کند.

|       |       |         |
|-------|-------|---------|
| $n$   | $m$   | $K$     |
| $a_1$ | $a_2$ | $\dots$ |
| $a_n$ |       |         |
| $p_2$ | $p_3$ | $\dots$ |
| $p_n$ |       |         |
| $v_1$ | $v_2$ | $\dots$ |
| $v_m$ |       |         |

## خروجی

دقیقاً  $m$  سطر چاپ کنید. در سطر  $z$ ام کمترین هزینه‌ی تأمین  $K$  کیلوگرم آرد در روز  $z$ ام را چاپ کنید؛ اگر موجودی کل شهر کمتر از  $K$  بود، عدد  $-1$  را چاپ کنید.

## محدودیت‌ها

$$2 \leq n \leq 3 \times 10^5$$

$$1 \leq m \leq 3 \times 10^5$$

$$1 \leq K \leq 10^{12}$$

$$1 \leq a_i \leq 10^6$$

$$1 \leq p_i < i$$

$$1 \leq v_j \leq n$$

## زیرمسئله‌ها

| محدودیت‌ها          | نمره | زیرمسئله |
|---------------------|------|----------|
| ورودی و خروجی نمونه | ۰    | ۱        |
| $a_i = 1$           | ۴    | ۲        |
| $v_j = 1$           | ۷    | ۳        |
| $K = 1$             | ۳    | ۴        |
| $K \leq 20$         | ۱۱   | ۵        |
| $n, m \leq 3000$    | ۱۲   | ۶        |
| بدون محدودیت اضافی  | ۶۳   | ۷        |

## ورودی و خروجی نمونه

| ورودی استاندارد | خروجی استاندارد |
|-----------------|-----------------|
| 3 4 4           | -1              |
| 5 2 7           | 16              |
| 1 1             | 16              |
| 2 1 3 2         | 11              |
| 4 3 3           | -1              |
| 4 1 1 10        | 12              |
| 1 1 3           | 3               |
| 3 2 1           |                 |

## توضیح نمونه‌ی اول:

- روز ۱: در کل شهر فقط ۱ کیلوگرم آرد تولید شده، اما به ۴ کیلوگرم آرد نیاز است. پس پاسخ ۱- است.
- روز ۲: موجودی نانوائی‌های ۱ و ۲ و ۳ به ترتیب ۱، ۲ و ۱ کیلوگرم است (مجموعاً ۴ کیلوگرم). قناد مجبور است تمام آرد موجود را بخرد که هزینه‌ی آن برابر  $16 = 1 \times a_1 + 2 \times a_2 + 1 \times a_3$  می‌شود.
- روز ۴: انبارها پرتر شده‌اند. حالا می‌توان ۳ کیلوگرم از نانوائی ارزان‌تر (نانوائی ۲ با قیمت ۲) و ۱ کیلوگرم از نانوائی ۱ (با قیمت ۵) تهیه کرد تا مجموع ۴ کیلوگرم با هزینه‌ی  $11 = 3 \times 2 + 1 \times 5$  تأمین شود.