مرتبش کن 1

لیست پیوندی یکطرفه‌ای را پیاده‌سازی کرده و با استفاده از آن کدی بنوسید که لیست پیوندی مرتب شده‌ای(غیر نزولی) را دریافت کرده که در آن یک عضو درجای اشتباهی قرار گرفته است آن را یافته و درجای درست قرار دهد.

▼ نکته‌ی بسیار مهم

- برای حل سوال باید جای گره‌ها را تغییر بدهید، صرفا تغییر مقدار داخل گره قابل قبول نیست.

مثال

ورودی نمونه ۱

```
1 1 2 3 4 5 4 6
```

خروجی نمونه ۱

```
1 1 2 3 4 4 5 6
```

مرتبش کن 2

دو لیست از اعداد داریم که اولی مرتب شده و دومی نامرتب است. میخواهیم **تنها با یک بار پیمایش لیست** اول همه عناصر لیست دوم را در لیست اول درج کنیم بصورتی که همچنان مرتب شده(غیرنزولی) باقی بماند و در خروجی هم همه‌ی اعداد را به ترتیب از روی لیست اول چاپ کنیم.

ورودی

دو خط از اعداد صحیح در ورودی دریافت میکنید که به ترتیب خط اول اعضای لیست اول بصورت سورت شده (غیرنزولی) و خط دوم شامل اعضای لیست دوم (بصورت نامرتب) می‌باشد.

$$0 \leq List.length \leq 10000$$

خروجی

خروجی باید شامل 1 خط باشد که ترتیب قرارگیری اعضای دو لیست در کنار یکدیگر است.

مثال

ورودی نمونه

```
1 10 11 12 12 12 20 45 100
90 1 61 3 1 2
```

خروجی نمونه

```
1 1 1 2 3 10 11 12 12 12 20 45 61 90 100
```

نکته مهم: در حل این سوال تنها مجاز به استفاده از لیست‌پیوندی یک‌طرفه می‌باشید و در صورت استفاده از لیست‌پیوندی دوطرفه هیچ نمره‌ای کسب نخواهید کرد!!

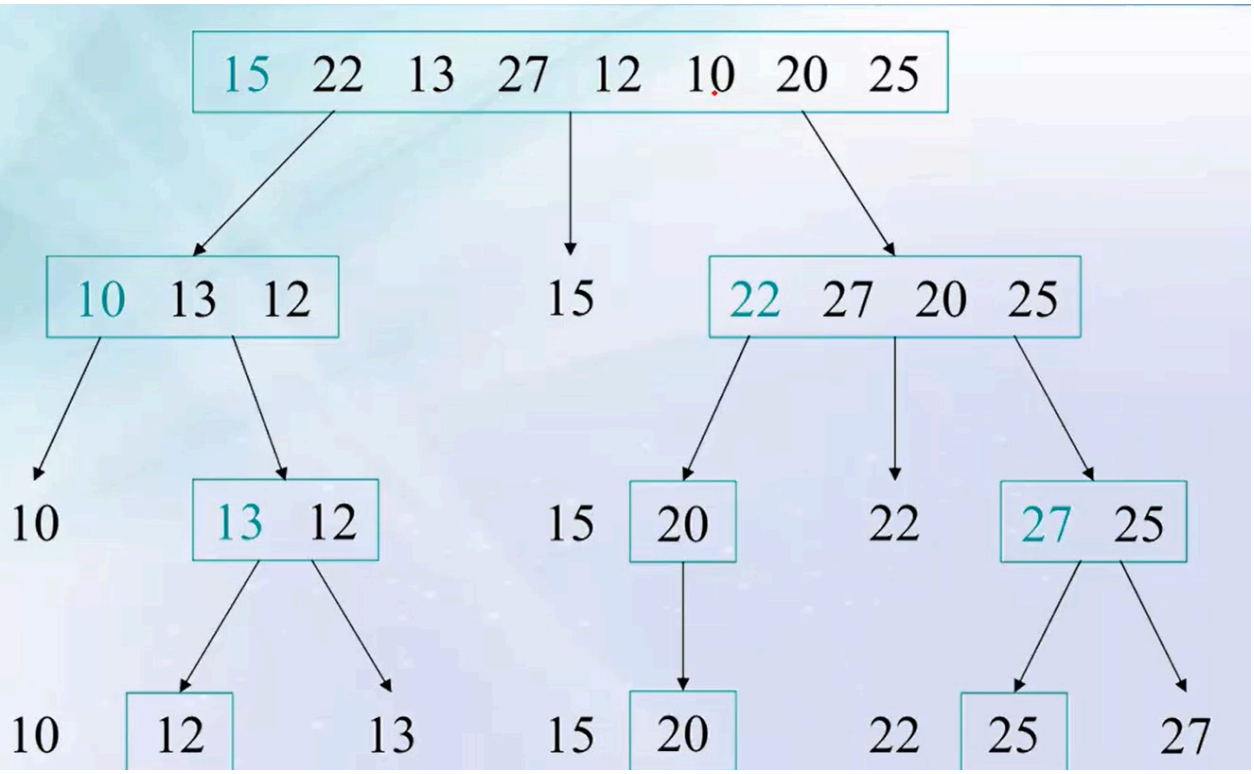
مرتبش کن3 - امتیازی

تا کنون با چند الگوریتم ـــمرتب‌سازی ـــدر آرایه ها (bubble sort, selection sort, insertion sort ..) آشنا شدید که این الگوریتم‌ها پیچیدگی نسبتا زیادی دارند. انواع الگوریتم‌های مرتب‌سازی دیگری هستند که پیچیدگی زمانی کمتری دارند که در ادامه با یکی از این الگوریتم‌ها آشنا می‌شویم.

در این الگوریتم با تقسیم مسئله به نمونه‌های کوچک‌تر و سپس ترکیب راه حل‌های نمونه‌های کوچک راه حل نمونه اصلی رابدست می‌آوریم.

مراحل مرتب‌سازی :

- انتخاب یک عنصر محوری (pivot, معمولا عنصر اول)
- افراز آرایه به دو بخش (partitioning) به طوری که عناصر کوچکتر از عنصر محوری در سمت چپ و عناصر بزرگتر در سمت راست آن قرار میگیرند.
- مرتب‌سازی هر بخش بصورت بازگشتی



تابع مرتب‌سازی :

```
void sort (index low,index high){
  index pivotpoint;
  if( high > low){
    partition(low, high,pivotpoint)

    sort(low, pivotpoint- 1)

    sort(pivotpoint + 1 ,high)
  }
}
```

فراخوانی این تابع در ابتدا بصورت sort(1,n) است.

تابع partition :

```
void partition (index low, index high, index& pivotpoint) {

  index i, j;
  keytype pivotitem;
  pivotitem = S[low];
  j = low;
  for (i = low + 1; i <= high; i++)

    if (S[i] < pivotitem) {
    //Choose first item for pivotitem
    j++;
    exchange S[i] and S[j];
    }

  pivotpoint = j;

  exchange S[low] and S[pivotpoint); //Put pivotitem at pivotpoint;
}
```

عملکرد این تابع : در هر تکرار اگر مشخص شود عنصری کوچکتر از عنصر محوری است به طرف چپ آرایه حرکت داده میشود.

شبه کدی بنویسید که با گرفتن یک لیست پیوندی آن را با استفاده از الگوریتم بالا بصورت صعودی مرتب کند. (دو تابع بالا باید با استفاده از لیست پیوندی پیاده‌سازی شوند.)

پالیندروم

در این معما تعداد n سوال به شما داده میشود که در هر سوال باید طول بزرگترین لیست پالیندروم را که در لیست پیوندی از اعداد داده شده رابیابید. سپس باید جواب هر قسمت را به هم وصل کنید تا کدی را بدست آورید البته باید چند نکته را هم رعایت کنید!

۱. حداکثر فضای اضافی استفاده شده قابل قبول برای حل یک پالیندروم برابر(O(1 میباشد.
۲. در این سوال یک عدد به تنهایی یک لیست پالیندروم به حساب نمی‌آید.
۳. برای حل این سوال باید از لینک لیست پیاده سازی شده توسط خود استفاده کنید.

ورودی

در خط اول عدد n که نشان دهنده تعداد سوالات است داده میشود سپس به تعداد n بار، هربار در خط اول m اندازه لیست پیوندی ورودی و در خط دوم به ترتیب اعداد موجود درلیست پیوندی داده میشود که اعدادلیست پیوندی به وسیله فاصله از هم جدا شده‌اند.

خروجی

شما باید یک عدد که بیانگر کد است را چاپ کنید.

مثال

ورودی نمونه ۱

```
3
8
4 3 2 1 2 3 5 3
6
1 2 1 1 2 1
10
1 2 3 1 2 3 1 2 3 4
```

خروجی نمونه ۱

```
560
```

همانطور که مشاهده میکنید، اولین ورودی داده شده حاوی 2 لیست پالیندروم {3 2 1 2 3}و {3 5 3}است که طول بزرگترین آنها برابر 5 میباشد.دومین ورودی حاوی 3 لیست پالیندروم {1 2 1} و {2 1 1 2} و {1 2 1 1 2 1} است که طول بزرگترین آنها برابر 6 میباشد.همچنین سومین ورودی حاوی لیست پالیندروم نیست پس بزرگترین طول آن 0 است که از کنار هم قرار دادن آنها عدد 560 بدست خواهد آمد و شما موفق به رمزگشایی قفل خواهید شد.

برعکسش کن

شبه کدی بنویسید که با در اختیار داشتن اعضای یک لیست پیوندی و دو عدد i و j عناصر بین دو عضوی که در جایگاه i ام تا j ام لیست پیوندی هستند را برعکس کند.

▼ نکته‌ی بسیار مهم

- برای حل سوال باید جای گره‌ها را تغییر بدهید، صرفا تغییر مقدار داخل گره قابل قبول نیست.

چرخشی1

شبه‌کدی بنویسید که دو لیست پیوندی چرخشی را به عنوان ورودی دریافت کند و حاصل اتصال `concatenation` دو لیست را در قالب یک لینک لیست پیوندی چرخشی برگرداند.

نکته مهم : فقط به عنصر head و از هر عنصر به عنصر بعدی آن دسترسی داریم.

▼ نمونه

Input: list1: 3 -> 2 -> 1, list2: 4 -> 5

Output: 3 -> 2 -> 1 -> 4 -> 5

چرخشی2

شبه‌کدی بنویسید که لیست‌پیوندی چرخشی با طول زوج را دریافت کرده و آنرا به دو لیست‌پیوندی چرخشی با طول یکسان تقسیم کند.

نکته مهم : فقط به عنصر head و از هر عنصر به عنصر بعدی آن دسترسی داریم.

▼ نمونه

Input: head: 13->3->8->9

Output: 13->3 , 8->9