

### سوال ۷ ساخت مثلث

$n$  پاره‌خط با طول‌های مثبت  $a_1, a_2, \dots, a_n$  داده شده است. الگوریتمی با مرتبه زمانی  $O(n \log n)$  ارائه دهید که تشخیص دهد آیا می‌توان سه پاره‌خط متمایز از میان آن‌ها انتخاب کرد که تشکیل مثلثی با مساحت مثبت دهند یا نه. در صورت وجود، یکی از سه‌تایی‌های معتبر را خروجی دهید؛ در غیر این صورت اعلام کنید که چنین سه‌تایی‌ای وجود ندارد.

اصل چهارم: برای سه طول به ترتیب صعودی:  $a \leq b \leq c$

در صورتی مثلث خواهیم داشت که:  $a + b > c$

حال برای ادامه از الگوریتم زیر استفاده می‌کنیم:

\* اگر در ابتدا طول‌ها را مرتب کنیم کافی هست تخم‌سای پشت سرهم را چک کنیم یعنی  $a[i] + a[i+1] > a[i+2]$  را با یک چک: چرا؟

$$\begin{matrix} x < y < k \\ \downarrow \\ a[x] \leq a[y] \leq a[k] \end{matrix}$$

$$a[x] + a[y] > a[k]$$

توجه کنید: اگر در سه‌تایی‌های غیر مجاور برقرار نباشد، در سه‌تایی‌های مجاور نیست: (دفعه خلف)

$$k > y+1 \Rightarrow a[y-1] + a[y] > a[k] \geq a[y+1] \Rightarrow *$$

$$x < y-1 \Rightarrow a[x] < a[y-1] \Rightarrow a[y-1] + a[y] > a[x] + a[y] > a[k] \geq a[y+1] \Rightarrow *$$

بنابراین روش هم کافی و لازم هست!

$$a[1] \leq a[2] \leq \dots \leq a[n-1] \leq a[n]$$

اگر یافت شد، همان سه‌تایی جواب است.

حال با الگوریتم‌های sort مثل merge sort در  $O(n \log n)$  طول‌ها را مرتب می‌کنیم:

برای  $i=1$  تا  $i=n-2$  پیمایش می‌کنیم و هر جا  $a[i] + a[i+1] > a[i+2]$  برقرار بود یک مثلث معتبر تشکیل می‌شود. اگر تا انتها پیدا نشد، اعلام عدم وجود.

$$O(n \log n) + O(n) \Rightarrow O(n \log n)$$

که در  $O(n)$  انجام می‌شود.

پیاده سازی :

```
Triangle(a[1..n]):  
  sort(a)  $\rightarrow O(n \log n)$   
  for i = 1 to n-2:  $\rightarrow O(n)$   
    if a[i] + a[i+1] > a[i+2]:  
      return (a[i], a[i+1], a[i+2])  
  return "No triangle exists"
```