



## آنالیز الگوریتم (۲۲۸۹۱)

مدرس: حسین بومری

[پاییز ۹۹]

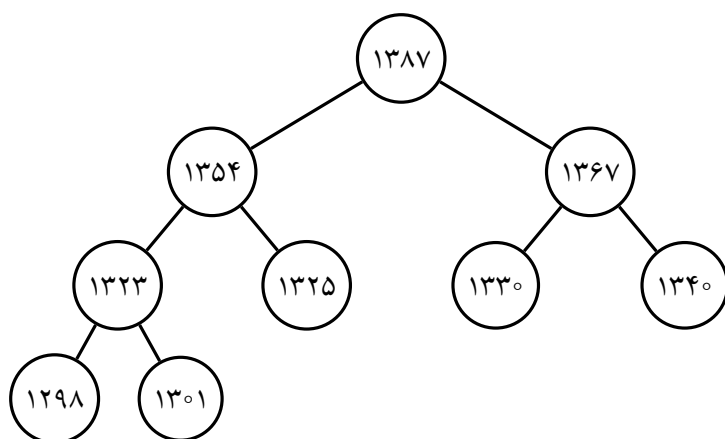
نگارنده: ۹۷۱۰۰۵۷۹ فرزین نصیری

تمرین ۱: تقسیم و حل، حریصانه و مرور ساختمان داده

### شجره نامه

با توجه به اینکه در هر گره سال تولد فردی که در آن است وجود دارد، همچنین با توجه به راست چین بودن و روند نزولی اعداد، ساختار شجره نامه در واقع یک Heap Max [۱] است.

یک مثال از شکل شجره به صورت زیر میتواند باشد:



هدف این است که با داشتن عدد  $x$ ، الگوریتمی ارائه دهیم که در زمان  $O(k)$  نشان آیا حداقل  $k$  گره وجود دارند بطوریکه سال هر گره بزرگتر از  $x$  باشد.

## الگوریتم ۱. الگوریتم به صورت زیر است:

```
main()
۱ // Driver code
۲ counter = ۰
۳ return k <= SearchTree(root, x, k, counter)

SearchTree(Node : node, Int : x, Int : k, Int : counter)
۱ if node.birth >= x
۲     counter ++
۳     if counter >= k
۴         return counter
۵     SearchTree(node.left, x, k, counter)
۶     SearchTree(node.right, x, k, counter)
۷ else // Do nothing !
```

ریشه درخت را  $root$  می‌نامیم که همان لیغراست. برای هر گره یک مقدار  $birth$  نگه‌میداریم و شروع به پیمایش میکنیم. اگر به گره‌ای برخورد کردیم که از سال  $x$  به بعد به دنیا آمده بود،  $counter$  را اضافه میکنیم. دقت کنید که نیازی به پیمایش کل درخت نیست که پایین‌تر اثبات می‌شود.

اثبات. اثبات درستی الگوریتم به صورت زیر است:

روش پیمایش درخت شجره نامه به این صورت است که به ازای هر گره، بررسی میکنیم که آیا زمان تولد آن شخص، بعد حد  $x$  است یا خیر. دقت کنید اگر به گره‌ای برسیم که قبل از  $x$  متولد شده است، با توجه به ویژگی شجره نامه یا همان  $heap, max$  میدانیم که اجداد آن شخص! (child nodes) حتماً زودتر به دنیا آمده‌اند در نتیجه نیاز به بررسی مجدد ندارند. در نتیجه  $counter$  فقط برای گره‌هایی که شرایط مسئله را دارند افزایش پیدا میکند.

□

اثبات. اثبات زمان الگوریتم به صورت زیر است:

برای پیدا کردن زمان الگوریتم باید تعداد گره‌هایی که الگوریتم پیمایش میکند تا به جواب برسد را پیدا کنیم. میدانیم که وقتی به  $k$  امین گره‌ی بزرگتر از  $x$  برسیم کار ما تمام شده است و اگر به این تعداد موجود نباشد گره‌های کمتری که مقدار بزرگتر از  $x$  داشتند را پیمایش کردیم. در نتیجه برای گره‌هایی که مقدار ناکمتر از  $x$  دارند، حداکثر  $k$  تا را بررسی میکنیم  $O(k)$ .

حال باید تعداد گره‌های پیمایش شده که مقادیر کمتری از  $x$  دارند را بشماریم. میدانیم که فرزندان گره‌ای که کمتری از  $x$  است نیز کمتر از  $x$  هستند و الگوریتم آن‌ها را بررسی نمی‌کند (خط ۷ کد جستجو). از طرفی اگر به گره‌ای برسیم که مقداری کمتر از  $x$  دارد، حتماً قبل از آن به گره‌ای رسیده بودیم که مقادیر بیشتر مساوی  $x$  داشته است و چون تعداد این گره‌ها طبق استدلال قبلی حداکثر  $k$  بود، پس تعداد گره‌های پیمایش شده با مقدار کمتر از  $x$  نیز از  $O(k)$  است.

□

از ترکیب دو استدلال با زمان کل الگوریتم نیز  $O(k)$  است.

