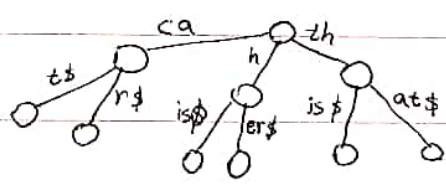



توی حالتی که node ها از برای بودن، اوردن حافظه، $O(|A| \sum w_i)$ میشه که خیلی بده. بیایم مثلاً اگر الفبا یون جویا تا بیت نشون می‌دایم، این t و k تیکه کنیم. بعد $\frac{t}{k}$ تیکه اول دهم و ... اینجوری که می‌دایم بعد توی خود درخت به من طبقه هم اضافه می‌شه ← $O(k \sum w_i |A|^k)$ می‌شه اوردن حافظه. انگار به Trie جدید با طول و الفبای جدید و ...

برجاش زمان $search / insert / del$... اینا اصلاً k بشن.

ایده‌ی دیگر: Compressed Trie: هر جا شاخه شاخه نیست، منسوخه کنیم.



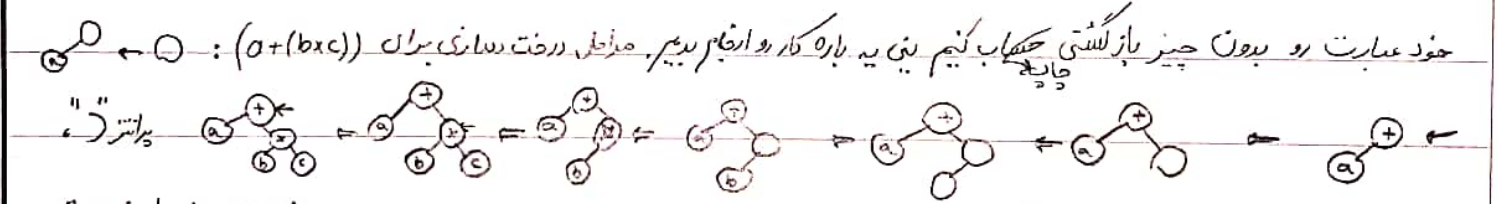
DS - جلسه ۱۱ - ۹۸, ۱, ۱۱ : درخت عبارت (عبارت) : $(c + a \times b)$ \rightarrow $add(c, mul(a, b)) = +c \times ba$

درختش :  ترتیب : محاسبه راست مقدار عبارت
 که این مدل میافزودنی نه مدل نمایی prefix (پیشوندی) infix

یا مثلا نمایی پسوندی : $cba \times +$! فایده پسوندی / پیشوندی به پرانتز نیاز ندارد ! (چرا؟) پرانتزها کار پردازش عبارت آسونتر میکنند. توصیف بازگشتی عبارت : $E \rightarrow (E \alpha E) \alpha = + - \times \dots$ (پرانتز) روی تبدیل این نماییها به هم کار کنیم مثلا تبدیل

درخت به عبارت چیزی بیایم pre / in / post order روی $E \rightarrow a$ متغیر $E \rightarrow (E \alpha E) \alpha = \log \sin \dots$ بیاید کنیم روی درخت و چاپ کنیم (محاسبه نمایی را فته) ریختل می الان داریم چاپ می کنیم عبارت با بدون نمایی که اول لقمیم. حالا تبدیل نمایی مثلا میافزودنی به درخت ... کافی از روی پرانتز بندی

هاستونیم اون α چاپ روی بیاید کنیم که $E \alpha E \alpha E$ که بتونیم بازگشتی بزنیم. α اولین چیز به جز پرانتز که تقاضا پرانتزها باز با سبزه $=$ نشود. (فرض \oplus ورودی پرانتز گذاری شده است) مثلا حالت BE ... متفرقا این از $O(n^2)$ در میاد ! کلا چاپ سبزه. ایده جایگزین :



```
for i=1 to n-1:
    if A[i] == '(' : p.left = new Node(); p = p.left
    else if A[i] == ')' : p = p.parent;
    else if operator O : p.key = O ; p.right = new Node; p = p.right
    else if operand v : p.key = v ; p = p.parent;
```

pointer رو یک باره بالا پرانتز " " یکی میاره پایین
 حالا روی تبدیل عبارت پیشوندی به درخت عبارت

کار کنیم. ایده کلی سن اینه \leftarrow به طرکی و به شرط اینکه $A[i]$ محل باشه دره خودی باره تابع که هم اندیس باره هم درخت ... ؟

```
Prefix To Tree (A, i)
t = new Node (A[i])
t.left, j = prefix to Tree (A, i+1)
t.right, k = prefix to Tree (A, j+1)
return t, k+1
```

اینجا خوبی recursive نش اینه که $O(n)$ در میاد ... در حالت n تغییر
 recursive هم با $O(n)$ میشه نوشت.

