

درخت عبارت

Expression Tree

مقدمه

- تا الان ما لیست را گسترش داده و درخت را تعریف کردیم.

- آیا درخت قدرتی به ما می‌دهد؟

- الگوریتمی برای حل معادلات ریاضی ارائه دهید؟

$$(8 - 5) \times (4 + 2) / 3$$

سوال

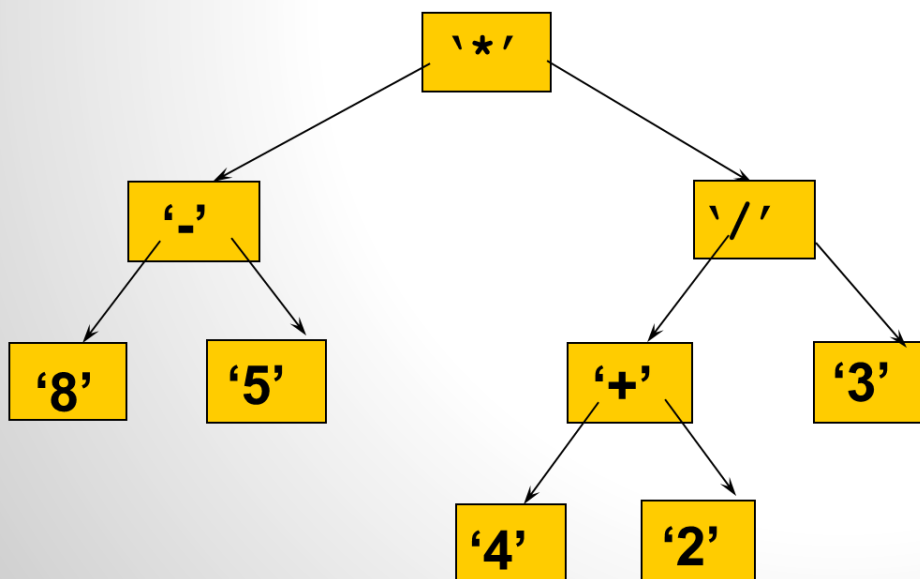
• الگوریتمی برای حل معادلات ریاضی ارائه دهید؟

$$(8 - 5) \times (4 + 2) / 3$$

• روش

1. ساخت **درخت عبارت**

2. حل به صورت بازگشتی



عبارت

- **عبارت:** مجموعه‌ای از عملگرها (operator) و عملوندها (operand) که با نظمی خاص یک مقدار را نشان می‌دهند.

$$(a + 5)$$

عبارت

- عبارت: مجموعه‌ای از عملگرها (operator) و عملوندها (operand) که با نظمی خاص یک مقدار را نشان می‌دهند.
- **عملگر**: یک سنبیل که یک عمل خاص را انجام می‌دهد.

$$(a + 5)$$

عبارت

- عبارت: مجموعه‌ای از عملگرها (operator) و عملوندها (operand) که با نظمی خاص یک مقدار را نشان می‌دهند.
- عملگر: یک سنبُل که یک عمل خاص را انجام می‌دهد.
- **عملوند**: یک مقدار است که عملیات عملگر بر روی آن انجام می‌شود.

$$(a + 5)$$

نمونه‌ای از یک عبارت

• یک عبارت پیچیده

$$f(a) + \sum_{k=1}^n \frac{1}{k!} \frac{d^k}{dt^k} \bigg|_{t=0} f(u(t)) + \int_0^1 \frac{(1-t)^n}{n!} \frac{d^{n+1}}{dt^{n+1}} f(u(t)) dt.$$

عبارت جبری

- عملیات جبری: نوعی از عملیات ریاضی می باشد که روی **یک عملوند (عملیات یگانی (unary))** یا روی **دو عملوند (عملیات دودویی)** انجام می شود.
- مثالی از عملیات جبری:

عملیات یگانی	عملیات باینری

- سوال: مقدار $5 - 2$ چقدر است؟

عبارت جبری

• یک عبارت جبری می‌تواند شامل چندین عملیات جبری باشد:

$$6 \times 3 + 2^4$$

• سوال مقدار بالا چقدر است؟

عبارت جبری

• یک عبارت جبری می‌تواند شامل چندین عملیات جبری باشد:

$$6 \times 3 + 2^4$$

• سوال مقدار بالا چقدر است؟

• ترتیب عملیات جبری: در ریاضیات هر عملیات اولویت مختلفی دارند.

• می‌توان این اولویت را با پرانتز نمایش داد.

$$((6 \times 3) + (2^4))$$

Context-free Grammar (CFG)

- در درس نظریه زبان تدریس خواهد شد:

- پس به صورت غیر رسمی: روشی است برای تولید کلمات یک زبان با گرامر مشخص

- در زمان فارسی:

- من به دانشگاه میروم.

- از دانشگاه را در فردا.

- برای جلوگیری از پیچیده شدن وارد جزئیات نمی‌شویم

Context-free Grammar (CFG)

• ساده ترین عبارت جبری:

a

2

• $E \rightarrow operand$

Context-free Grammar (CFG)

• عبارات یگانی (unary):

(-2)

$(!(a))$

$(\sin \theta)$

• $E \rightarrow operand$

• $E \rightarrow (\alpha E)$

• $\alpha \rightarrow - \mid ! \mid \sin \mid \log \mid \dots$

Context-free Grammar (CFG)

• عبارات دودویی:

$$(2 + b)$$

$$(c^{(8 - 4)})$$

$$(True \wedge S)$$

$$\bullet E \rightarrow operand$$

$$\bullet E \rightarrow (\alpha E)$$

$$\bullet \alpha \rightarrow - \mid ! \mid \sin \mid \log \mid \dots$$

$$\bullet E \rightarrow (E\beta E)$$

$$\bullet \beta \rightarrow + \mid \times \mid ^ \mid \wedge \mid \dots$$

Context-free Grammar (CFG)

- $E \rightarrow \textit{operand}$
- $E \rightarrow (\alpha E)$
- $\alpha \rightarrow - \mid ! \mid \sin \mid \log \mid \dots$
- $E \rightarrow (E\beta E)$
- $\beta \rightarrow + \mid \times \mid ^ \mid \wedge \mid \dots$
-

مثال ١:

$((8 - 5) \times ((4 + 2)/3))$

Context-free Grammar (CFG)

- $E \rightarrow operand$
- $E \rightarrow (\alpha E)$
- $\alpha \rightarrow - \mid ! \mid \sin \mid \log \mid \dots$
- $E \rightarrow (E\beta E)$
- $\beta \rightarrow + \mid \times \mid ^ \mid \wedge \mid \dots$
-

$(a + \times)$

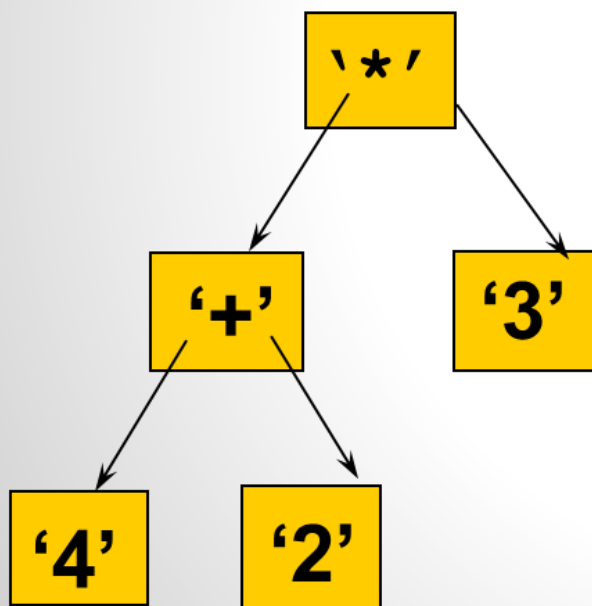
$(a($

$+$

مثال ٢:

درخت عبارت

- درخت عبارت: نوعی درخت دودویی برای نمایش یک عبارت است.
- چون عملگرها یگانی (unary) یا دودویی هستند



درخت عبارت

- ساده ترین عبارت جبری:

$E \rightarrow operand$

- معادل یک برگ با مقدار عملوند.



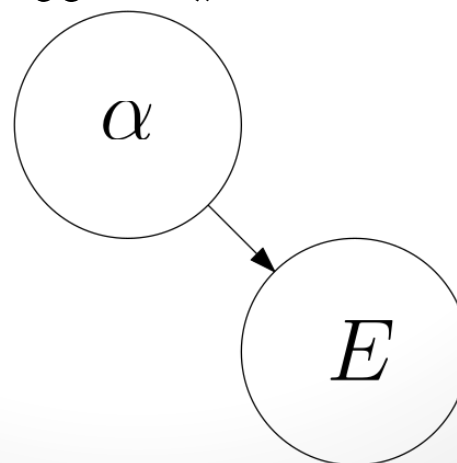
درخت عبارت

- عبارات یگانی (unary):

$$E \rightarrow (\alpha E)$$

$$\alpha \rightarrow - \mid ! \mid \sin \mid \log \dots$$

- معدل یک گره میانی می‌باشند که فقط یک فرزند راست (زیر درخت) دارند.



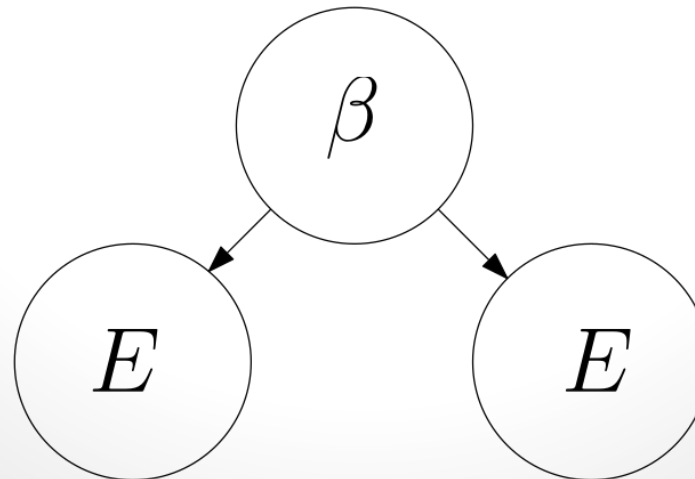
درخت عبارت

- عبارات دودویی:

$$E \rightarrow (E\beta E)$$

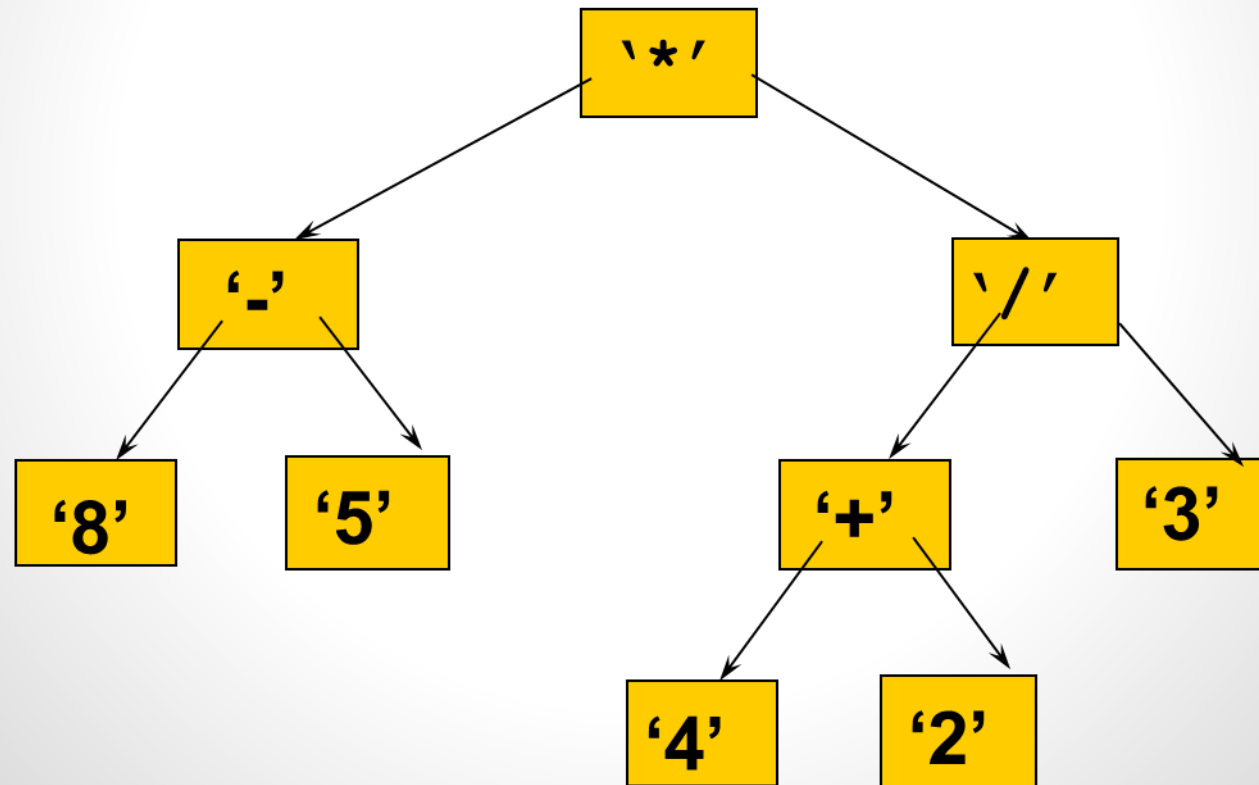
$$\beta \rightarrow + | \times | ^ | \wedge | \dots$$

- معدل یک گره میانی می‌باشند که دو فرزند (زیر درخت) دارند.



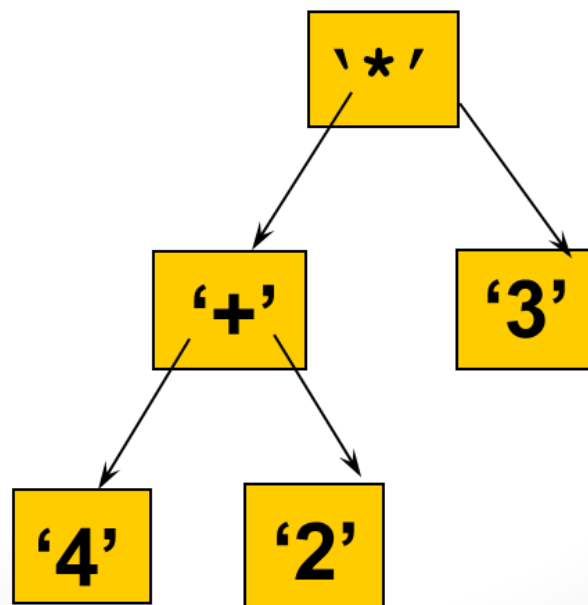
مثال

• درخت عبارت $((8 - 5) \times ((4 + 2)/3))$



مقداریابی

- نحوه پیدا کردن مقدار نهایی یک عبارت با استفاده از درخت عبارت (به صورت بازگشتی)



نگارش میانوندی (infix)

• نگارش عبارت به صورتی که عملگر میان عملوند یا عبارات زیرین می آید:

$$2 + a$$

$$((A \times (B + C)) / D)$$

- $E \rightarrow operand$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E \beta E)$

نیاز به پرانتزبندی کامل یا دانستن اولویتها دارد.

معدل پیمایش inorder درخت عبارت

نگارش پسوندی (postfix)

• نگارش عبارت به صورتی که عملگر پس از عملوندها یا عبارات زیرین می آید:

$2a+$

ABC+xD/

- $E \rightarrow operand$
- $E \rightarrow E\alpha$
- $E \rightarrow EE\beta$

نیاز به پرانتزبندی ندارد

معدل پیمایش postorder درخت عبارت

نگارش پیشوندی (prefix)

• نگارش عبارت به صورتی که عملگر پیش از عملوندها یا عبارات زیرین می آید:

$+2a$

$/\times A + BCD$

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow \beta EE$

نیاز به پرانتزبندی ندارد

معدل پیمایش preorder درخت عبارت

مسئله ۱

- ساخت درخت عبارت با استفاده از نگارش میانوندی با پرانتزبندی کامل:

$$((A \times (B + C)) / D)$$

پسوندی:

$$ABC + \times D /$$

پیشوندی:

$$/ \times A + BCD$$

ساخت درخت عبارت از نگارش میانوندی

پیدا کردن ریشه:

- $E \rightarrow operand$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E\beta E)$

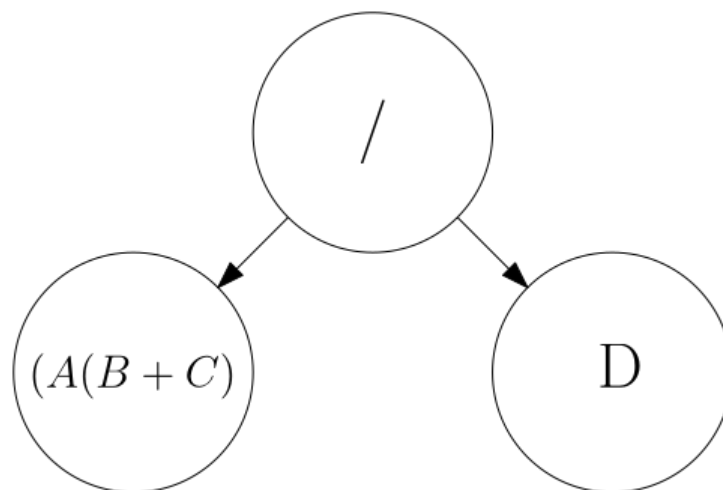
در کدام یک از دستورات فوق قرار دارد؟

$$((A \times (B + C)) / D)$$

ساخت درخت عبارت از نگارش میانوندی

پیدا کردن ریشه:

- $E \rightarrow operand$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E\beta E)$



پیدا کردن هر بخش با شمارش پرانتز ها

$((A \times (B + C)) / D)$

ساخت درخت عبارت از نگارش میانوندی

پیدا کردن ریشه:

- $E \rightarrow operand$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E\beta E)$

حل کردن به صورت بازگشتی برای هر زیر بخش

$((A \times (B + C)) / D)$

ساخت درخت عبارت از نگارش میانوندی

• زمان اجرا:

$$((((a + b) + c) + d) + e) + f)$$

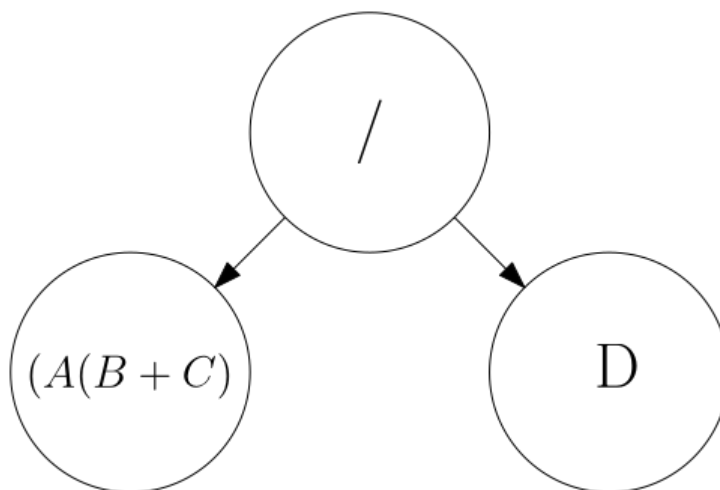
پس زمان اجرا: $O(n^2)$

پیاده سازی در زمان خطی؟

ساخت درخت عبارت از نگارش میانوندی

پیدا کردن ریشه:

- $E \rightarrow (operand)$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E\beta E)$



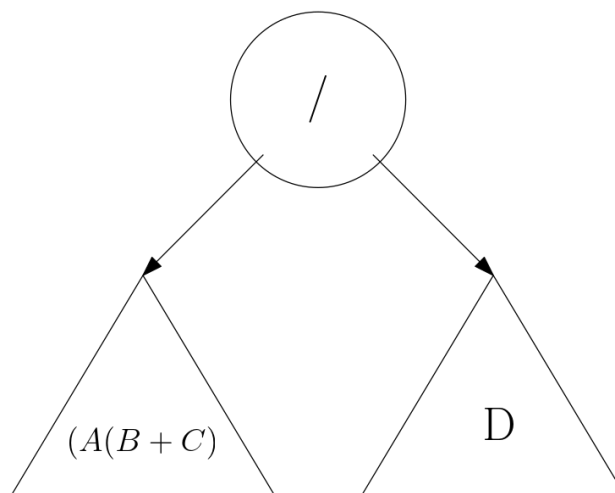
بجای پیدا کردن هر بخش

$((A \times (B + C)) / D)$

ساخت درخت عبارت از نگارش میانوندی

پیدا کردن ریشه:

- $E \rightarrow (operand)$
- $E \rightarrow (\alpha E)$
- $E \rightarrow (E\beta E)$



هنگام پیدا کردن هر بخش

زیر درخت مربوطه را می‌سازیم

$((A \times (B + C)) / D)$

مسئله ۱

- ساخت درخت عبارت با استفاده از نگارش پیشوندی و پسوندی

پسوندی:

$ABC+xD/$

پیشوندی:

$/xA+BCD$

پیدا کردن ریشه

پیشوندی:

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow \beta EE$

$/xA+BCD$

پسوندی:

- $E \rightarrow operand$
- $E \rightarrow E\alpha$
- $E \rightarrow EE\beta$

$ABC+xD/$

پیدا کردن عبارت اول

پیشوندی:

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow \beta E E$

پسوندی:

- $E \rightarrow operand$
- $E \rightarrow E \alpha$
- $E \rightarrow E E \beta$

ABC+xD/

/xA+BCD

پیدا کردن عبارت دوم

پیشوندی:

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow \beta E E$

پسوندی:

- $E \rightarrow operand$
- $E \rightarrow E \alpha$
- $E \rightarrow E E \beta$

ABC+xD/

/xA+BCD

مسئله ۲

- ساخت درخت عبارت با استفاده از نگارش میانوندی (بدون پرانتز)

مسئله ۲

- مدل ساده شده (فقط جمع و تفریق و ضرب و تقسیم و توان):

$$E \rightarrow operand \quad \bullet$$

$$E \rightarrow \alpha E \quad \bullet$$

$$\alpha \rightarrow - \quad \bullet$$

$$E \rightarrow E\beta E \quad \bullet$$

$$\beta \rightarrow + \mid - \mid \times \mid \div \mid ^ \quad \bullet$$

- اولویت عملگرها اول با توان است بعد با ضرب و تقسیم و در نهایت با جمع و تفریق
- اگر اولویت ها یکسان باشد، از سمت چپ محاسبه می‌کنیم.

مسئله ۲

• روش ساخت درخت: به صورت مرحله‌ای درخت را می‌سازیم:

$$a + b \times c \div d - e$$

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

مسئله ۲

• a . یک عملوند است، پس یک گره برگ برای آن می‌سازیم

$$a + b \times c \div d - e$$

- $E \rightarrow \text{operand}$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

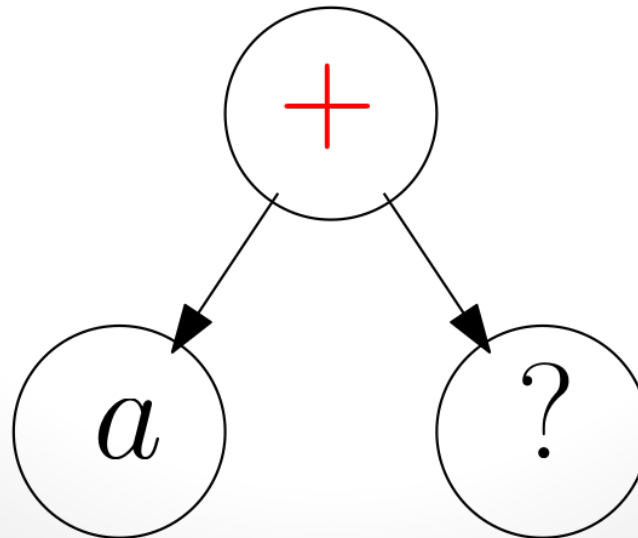


مسئله ۲

- جمع یک عملگر دودویی است پس آیا می‌توان گره میانی مربوطه را ساخت؟
- فرزندانش کدام اند؟

$$a + b \times c \div d - e$$

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$



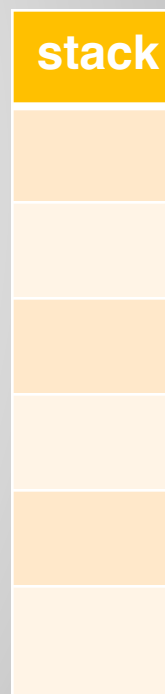
مسئله ۲

• از انجایی که تکلیف جمع مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

a

$a + b \times c \div$



مسئله ۲

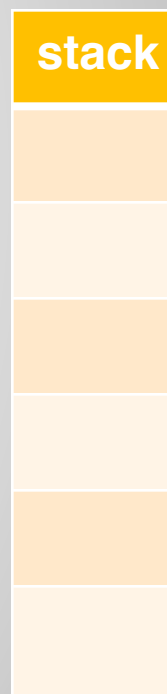
• b . یک عملوند است، پس یک گره برگ برای آن می‌سازیم

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

a

b

$a + b \times c \div$

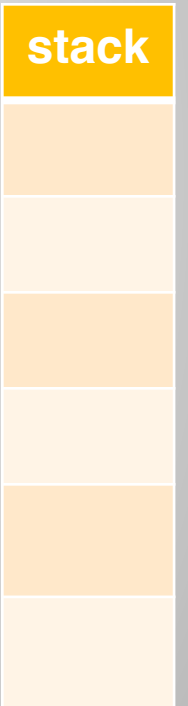


مسئله ۲

• از انجایی که تکلیف ضرب مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

$a + b \times c \div$



چون اولویت ضرب بیشتر از جمع است، تکلیف جمع مشخص نمی‌شود

مسئله ۲

• c . یک عملوند است، پس یک گره برگ برای آن می‌سازیم

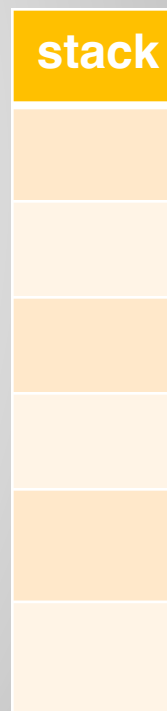
- $E \rightarrow \textit{operand}$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

a

b

c

$a + b \times c \div$



مسئله ۲

• از انجایی که تکلیف تقسیم مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

$a + b \times c \div$

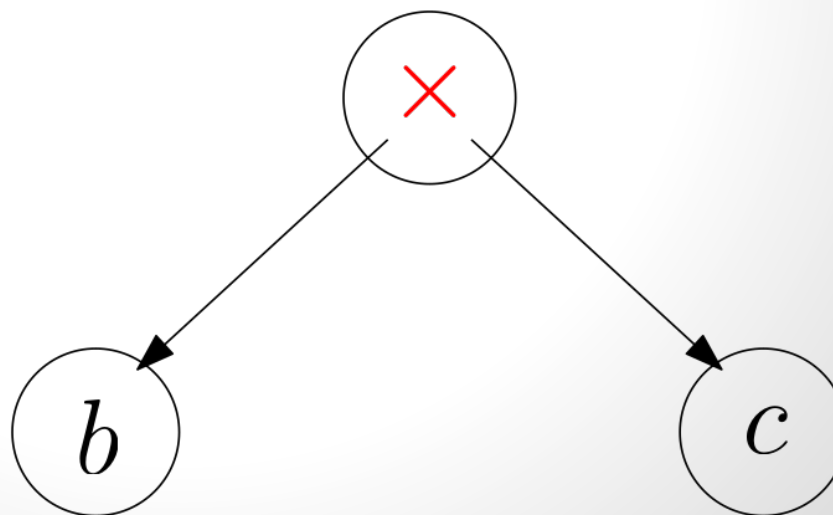
stack

چون اولویت تقسیم برابر ضرب است و اولویت با چپ‌ترین است، تکلیف
رابطه مشخص می‌شود

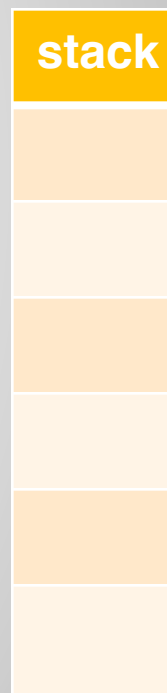
مسئله ۲

• دو زیردرخت مرحله قبل فرزندان ضرب می‌شوند.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$



$a + b \times c \div$

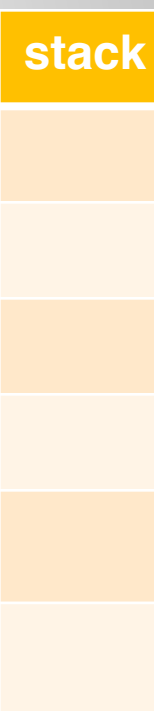


مسئله ۲

• از انجایی که تکلیف تقسیم مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

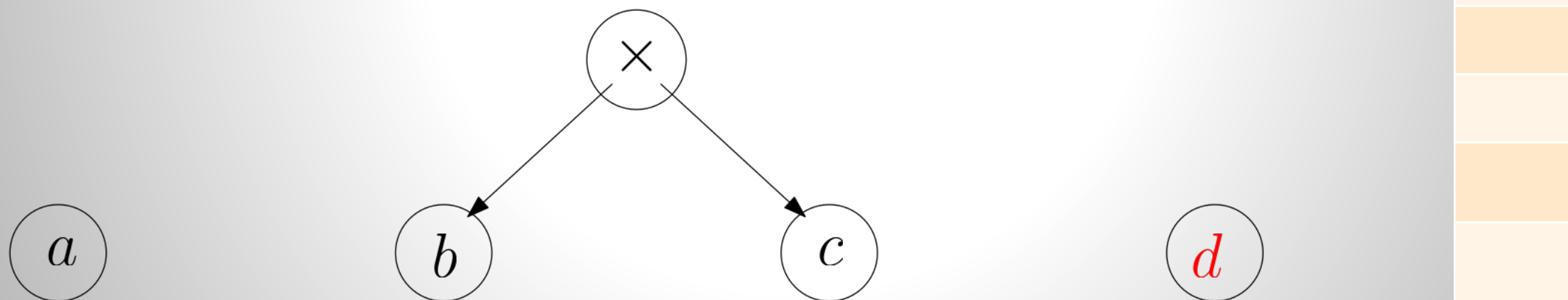
$$a + b \times c \div$$



چون اولویت تقسیم بیشتر از جمع است، تکلیف جمع مشخص نمی‌شود

مسئله ۲

• d . یک عملوند است، پس یک گره برگ برای آن می‌سازیم

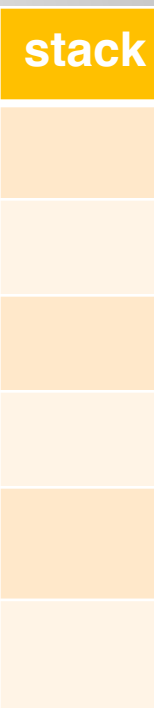


مسئله ۲

• از انجایی که تکلیف تفریق مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

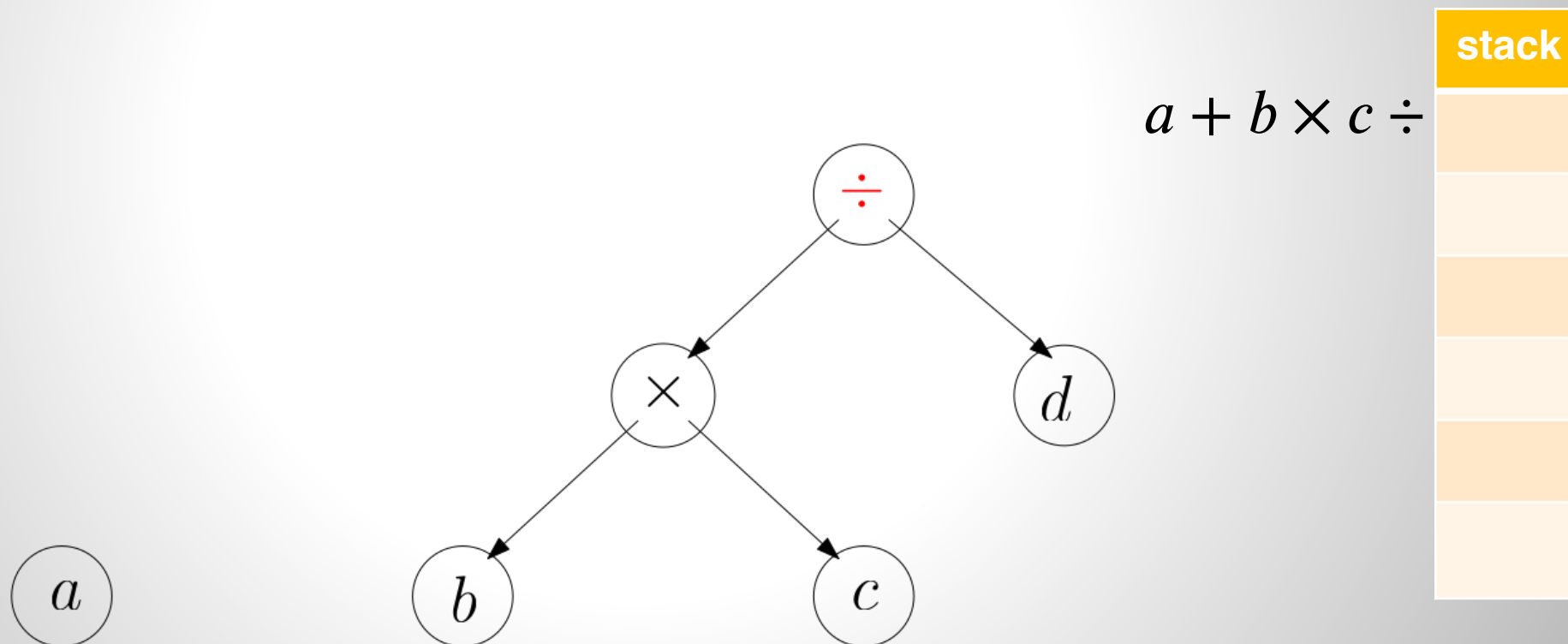
$a + b \times c \div$



چون اولویت تقسیم از تفریق بیشتر است

مسئله ۲

- دو زیردرخت مرحله قبل فرزندان تقسیم می‌شوند.



مسئله ۲

• از انجایی که تکلیف تفریق مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

$a + b \times c \div$

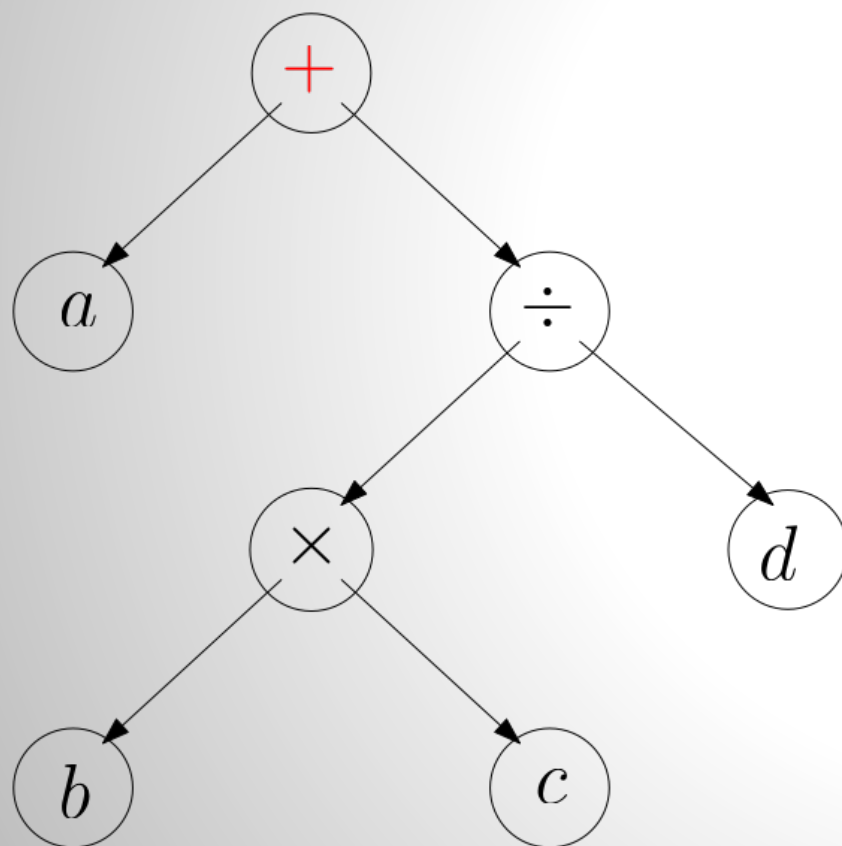
stack

چون اولویت تفریق برابر جمع است و اولویت با چپ‌ترین است، تکلیف جمع

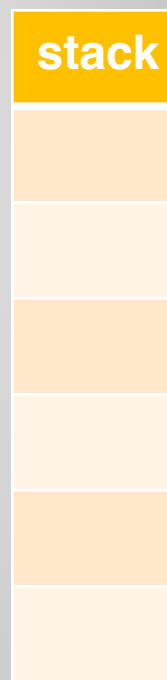
مسحک می‌شود

مسئله ۲

- دو زیردرخت مرحله قبل فرزندان جمع می‌شوند.



$a + b \times c \div$

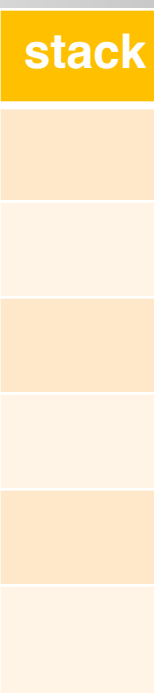


مسئله ۲

• از انجایی که تکلیف تفریق مشخص نیست آن را به یک پشته اضافه می‌کنیم.

- $E \rightarrow operand$
- $E \rightarrow \alpha E$
- $E \rightarrow E\beta E$

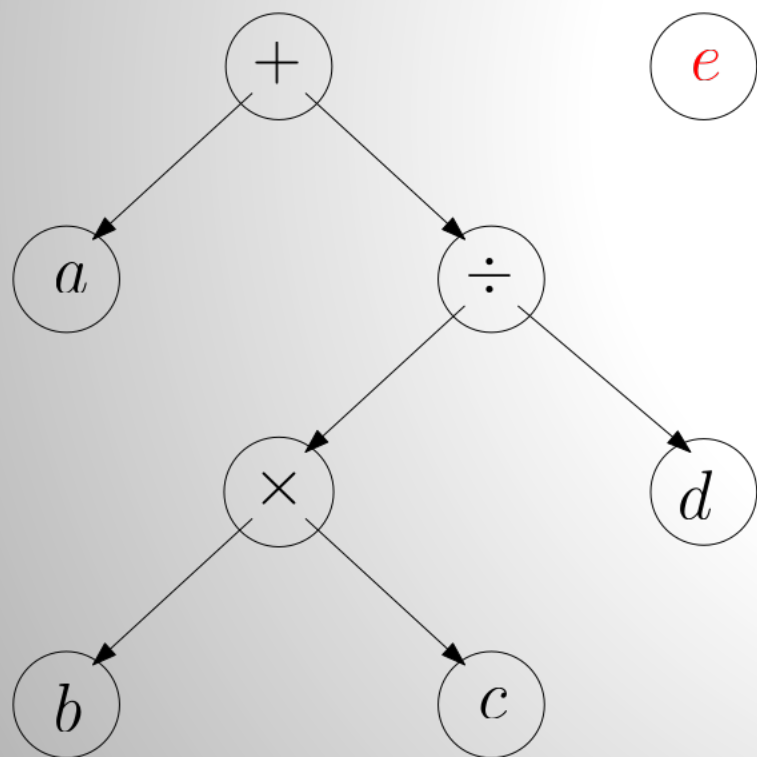
$a + b \times c \div$



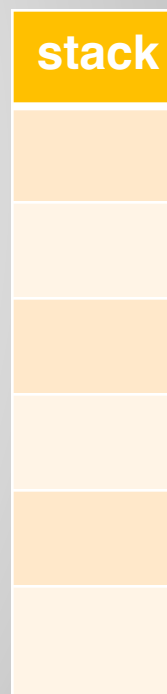
چون پشته خالی شد

مسئله ۲

• e . یک عملوند است، پس یک گره برگ برای آن می‌سازیم

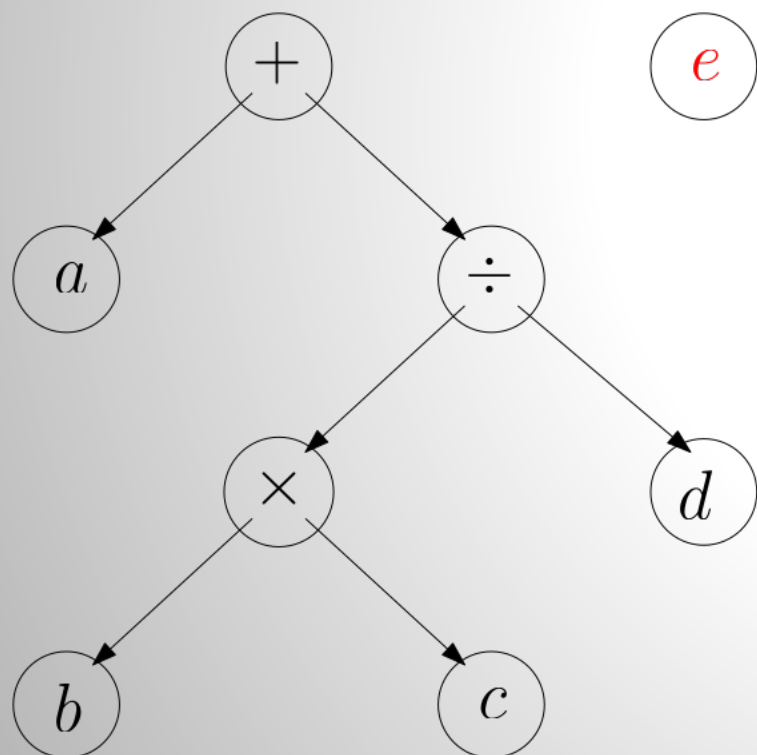


$a + b \times c \div$



مسئله ۲

- چون عبارت تمام شد، تکلیف پشته معلوم است و دو زیردرخت مرحله قبل فرزندان تفکیک می‌شوند.



$$a + b \times c \div$$

stack

مسئله ۲

