Dynamic Programming: Placing Parentheses

Alexander S. Kulikov

St. Petersburg Department of Steklov Institute of Mathematics
Russian Academy of Sciences

Data Structures and Algorithms
Algorithmic Toolbox

Outline

How to place parentheses in an expression

$$1 + 2 - 3 \times 4 - 5$$

to maximize its value?

How to place parentheses in an expression

$$1 + 2 - 3 \times 4 - 5$$

to maximize its value?

Example

- $((((1 + 2) - 3) \times 4) - 5) = -5$

How to place parentheses in an expression

$$1 + 2 - 3 \times 4 - 5$$

to maximize its value?

Example

- $((((1 + 2) - 3) \times 4) - 5) = -5$
- $((1 + 2) - ((3 \times 4) - 5)) = -4$

Answer

$$((1 + 2) - (3 \times (4 - 5))) = 6$$

Another example

What about

$$5 - 8 + 7 \times 4 - 8 + 9\,?$$

Soon

We'll design an efficient dynamic programming algorithm to find the answer.

Outline

Placing parentheses

Input: A sequence of digits $d_1, \ldots, d_n$ and a sequence of operations $op_1, \ldots, op_{n-1} \in \{+, -, \times\}$.

Output: An order of applying these operations that maximizes the value of the expression

$$d_1 \, op_1 \, d_2 \, op_2 \cdots op_{n-1} \, d_n \, .$$

Intuition

- Assume that the last operation in an optimal parenthesizing of $5 - 8 + 7 \times 4 - 8 + 9$ is $\times$:

$$(5 - 8 + 7) \times (4 - 8 + 9) .$$

- It would help to know optimal values for subexpressions $5 - 8 + 7$ and $4 - 8 + 9$.

However

We need to keep track for both the minimal and the maximal values of subexpressions!

Example: $(5-8+7)\times(4-8+9)$

$$\begin{aligned}
\min(5-8+7) &= (5-(8+7)) = & -10\\
\max(5-8+7) &= ((5-8)+7) = & 4\\
\min(4-8+9) &= (4-(8+9)) = & -13\\
\max(4-8+9) &= ((4-8)+9) = & 5
\end{aligned}$$

$$\max((5-8+7)\times(4-8+9)) = 130$$

Subproblems

- Let $E_{i,j}$ be the subexpression

$$d_i \, op_i \cdots op_{j-1} \, d_j$$

- Subproblems:

$$M(i,j) = \text{maximum value of } E_{i,j}$$
$$m(i,j) = \text{minimum value of } E_{i,j}$$

Recurrence Relation

$$M(i,j) = \max_{i \le k \le j-1} \begin{cases} M(i,k) & op_k & M(k+1,j) \\ M(i,k) & op_k & m(k+1,j) \\ m(i,k) & op_k & M(k+1,j) \\ m(i,k) & op_k & m(k+1,j) \end{cases}$$

$$m(i,j) = \min_{i \le k \le j-1} \begin{cases} M(i,k) & op_k & M(k+1,j) \\ M(i,k) & op_k & m(k+1,j) \\ m(i,k) & op_k & M(k+1,j) \\ m(i,k) & op_k & m(k+1,j) \end{cases}$$

Outline

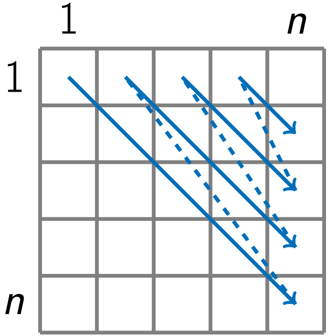
`MinAndMax`(i, j)

```
min ← +∞
max ← −∞
for k from i to j − 1:
    a ← M(i, k)  op_k  M(k + 1, j)
    b ← M(i, k)  op_k  m(k + 1, j)
    c ← m(i, k)  op_k  M(k + 1, j)
    d ← m(i, k)  op_k  m(k + 1, j)
    min ← min(min, a, b, c, d)
    max ← max(max, a, b, c, d)
return (min, max)
```

Order of Subproblems

- When computing $M(i, j)$, the values of $M(i, k)$ and $M(k + 1, j)$ should be already computed.
- Solve all subproblems in order of increasing $(j - i)$.

Possible Order



Parentheses($d_1\, op_1\, d_2\, op_2 \ldots d_n$)

```
for i from 1 to n:
   m(i, i) ← d_i ,  M(i, i) ← d_i
for s from 1 to n − 1:
   for i from 1 to n − s:
      j ← i + s
      m(i, j), M(i, j) ← MinAndMax(i, j)
return M(1, n)
```

Example: $5 - 8 + 7 \times 4 - 8 + 9$

5	-3	-10	-55	-63	-94
	8	15	36	-60	-195
		7	28	-28	-91
			4	-4	-13
				8	17
					9

m

5	-3	4	25	65	200
	8	15	60	52	75
		7	28	20	35
			4	-4	5
				8	17
					9

M

Outline

1. Problem Overview
2. Subproblems
3. Algorithm
4. Reconstructing a Solution

Example: $5 - 8 + 7 \times 4 - 8 + 9$

5	-3	-10	-55	-63	-94
	8	15	36	-60	-195
		7	28	-28	-91
			4	-4	-13
				8	17
					9

m

5	-3	4	25	65	200
	8	15	60	52	75
		7	28	20	35
			4	-4	5
				8	17
					9

M