1092 演算法 Homewoek 3

Algorithm Design

Data Structure

1. Dynamic Array

Algorithm

我使用類似 Matrix-Chain Multiplication 的演算法，把每個運算子當作 root，嘗試每個運算子當作 root，最後的結果拿得到最大值的結果。這個題目稍微有點變化，因為整數運算有正負號，所以每一格需要存最小值以及最大值，運算可能由：{最小值 op 最小值, 最小值 op 最大值, 最大值 op 最小值, 最大值 op 最大值} 組成。取得最大值後，括號的方式是遞迴，原本想紀錄來源路徑，但是寫到後面發現空間複雜度不理想，以及很難實作。所以我採取遞迴嘗試哪個結果組成目前這個值，因為遞迴想的太複雜，所以原本只要求最大值，不小心在同個遞迴函式裡連最小值的寫好了。

Pseudo Code

1. 區間最大值的 2 維陣列

```
interval[...][...].max = -INF
interval[...][...].min = INF
interval[x][x] = x

for w = 1 ... size(operators) + 1
  for w1 = 0 ... size(operators.size()) - w
    col = w + w1;
    for w2 = w1 ... col
      val = calc(
        interval[w1][w2],
        operators[w2],
        interval[w2 + 1][col]
      );
      interval[w1][col].min = min(interval[w1][col].min, val.min);
      interval[w1][col].max = max(interval[w1][col].max, val.max);
```

2. 取得最大值或最小值的括號運算式

```
def minmax_expression(start_left, end_right, bool get_min_or_max):
  if (start_left == end_right)
      return operands[start_left];
  elif (start_left + 1 == end_right)
    if (start_left < size(operators))
      return (
        "(" +
        operands[start_left] +
        operators[start_left] +
        operands[start_left + 1]) +
        ")"
      );
    else
      return operands[x];


  now = interval[x][y].(get_min_or_max)
  for w2 = start_left ... end_right + 1
    vals = {
      calc(
        interval[x][w2].min,
        operators[w2],
        interval[w2 + 1][y].min
      ),
      calc(
        interval[x][w2].min,
        operators[w2],
        interval[w2 + 1][y].max
      ),
      calc(
        interval[x][w2].max,
        operators[w2],
        interval[w2 + 1][y].min
      ),
      calc(
        interval[x][w2].max,
        operators[w2],
        interval[w2 + 1][y].max
      )
    }

    val_idx = max_direct ? index_of(max(vals)) : index_of(min(vals));

    if (max(vals) == now || min(vals) == now)
      return (
        "(" +
        minmax_expression(x, w2, now.left(), val_idx >= 2) +
        operators[w2] +
        minmax_expression(x, w2, now.right(), val_idx & 1) +
        ")"
      )

```

Analysis

1. 區間最大值的 2 維陣列
   n^2**區間範圍** $*$ n**嘗試每個當** $\text{root} = O(n^3)$
2. 取得最大值或最小值的括號運算式
   n**嘗試每個當** $\text{root} * log(n)$**每次分 2 塊** $= O(n \log(n))$