

Прошу прощение за отсутствие кода, нет времени чтобы это нормально закодить

## Задание 1

Дана правильная скобочная последовательность. Нужно найти у каждой скобки соответствующую пару за  $O(n)$  work и  $O(\sqrt{n} \text{ polylog } n)$  span.

1. Разбиваем строку на  $\sqrt{n}$  блоков

- Параллельно на каждом блоке  $k$  вычисляем пары, внутри блока алгоритм последовательный
- Сохраняем индексы скобочек без пар в массивы  $\text{open}[k]$ ,  $\text{close}[k]$

$$\text{work} = O(n), \text{scan} = O(\sqrt{n} \cdot \log \sqrt{n}) = O(\sqrt{n} \cdot \log n)$$

2. Дальше строим бинарное дерево

- В листьях храним указатели на массивы  $\text{open}[k]$ ,  $\text{close}[k]$
- В вершинах храним количество незаматченных открывающихся и закрывающихся скобок ( $\text{len}(\text{open})$ ,  $\text{len}(\text{close})$ ), а так же число пар, которые образуются  $\text{match} = \min(\text{len}(\text{left\_child.open}), \text{len}(\text{right\_child.close}))$

$$\text{work} = O(n), \text{scan} = O(\log \sqrt{n}) = O(\log n)$$

3. Получается по дереву из п.2 мы можем за  $\log \sqrt{n}$  найти пару для каждой оставшейся скобки

- Для  $\text{open}[i][j]$  ( $j$ -я открывающаяся скобка из  $i$ -го блока).
  - Поднимаемся по дереву вверх, пока сумма  $\text{match}$  пройденных вершин не привысит  $\text{len}(\text{open}[i]) - j$
  - Спускаемся по правому поддереву найденной вершины, ища закрывающуюся скобку с нужным индексом, мы можем это сделать так как каждая вершина хранит количество скобок без пар
- в худшем случае поиск п скобок, за  $\log \sqrt{n}$  каждая в `ParallelFor`, займет:

$$\text{work} = O(n), \text{scan} = O(\log n \cdot \log \sqrt{n}) = O(\log^2 n)$$

## Задание 2

Дано выражение, где каждая операция обрамлена скобками, а operandы - цифры. Постройте дерево вычислений за  $O(n)$  work и  $O(\sqrt{n} \text{ polylog } n)$  span.