Извиняюсь 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 скобок без пар в массивы $\text{open}[k]$, $\text{close}[k]$

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

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 также число пар, которые образуются $\text{match} = \min(\text{len}(\text{left\_child.open}), \text{len}(\text{right\_child.close}))$

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

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 match пройденных вершин не превышает $\text{len}(\text{open}[i]) - j$
     - Спускаемся по правому поддереву найденной вершины, ищя закрывающуюся скобку с нужным индексом, мы можем это сделать так как каждая вершина хранит количество скобок без пар
   - в худшем случае поиск n скобок, за $\log \sqrt{n}$ каждая в `ParallelFor`, займет:

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

Задание 2

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

Напишу идею, времени довести ее до ума нет :(

1. Ищем пары для скобок по алгоритму из 1 задания
2. Глубина вложенности для каждого символа
   - Сопоставляем `(` - +1, `)` - −1, остальные символы - 0
   - Делаем scan - получаем глубину вложенности
3. Зная глубину вложенности мы можем для каждой скобки определить соответсвтующий ей оператор, и, наоборот, для каждого оператора мы можем найти соответствующие скобки
4. Разбиваем строку на $\sqrt{n}$ блоков
   - Параллельно на обрабатываем блоки, внутри блока алгоритм последовательный
   - В блоке строим поддеревья, цифры в листья, операторы в вершины
5. Слияние блоков
   - Кажется, зная индексы соответствующих скобок, операторов и операндов можем параллельно сливать соседние блоки и соединять построенные в них поддеревья, уменьшая каждый раз количество блоков в 2 раза