Семинар #3

Антон Кухтичев

образование

Содержание занятия

- Сумма всех чисел от корня-до-листа (medium)
- Самая длинная подстрока с уникальными символами (medium)
- Сумма значений листьев на самом глубоком уровне (medium)
- Протухшие апельсины (medium)
- Максимальный элемент на каждом уровне (medium)
- Слияние k-списков (hard)

Сумма всех чисел от корня-до-листа (medium)

- Дан указатель на корень;
- Найти сумму, полученную сложением всех чисел, получаемых от корня до листа

```
int sumNumbers(TreeNode* root);
```

Ссылка:

https://interview.cups.online/live-coding/?room=389ccf33-6a3a-451d-b391-3bc52d51e1b3

Самая длинная подстрока с уникальными символами (medium)

- Дана строка str
- Найти самую длинную подстроку с уникальными символами

```
int lengthOfLongestSubstring(const std::string& str)
```

Ссылка:

https://interview.cups.online/live-coding/?room=9ce76c20-f016-4541-9788-06750c3fcd16

Сумма значений листьев на самом глубоком уровне (medium)

- Дан указатель на корень
- Найти сумму листьев на самом глубоком уровне

```
int deepestLeavesSum(TreeNode* root);
```

Ссылка:

https://interview.cups.online/live-coding/?room=f9d92df9-5003-45a0-a505-c1212a35796e

Протухшие апельсины (medium)

- Дана матрица n на m, где каждая ячейка может принимать одно из значений
  - 0 представляет пустую ячейку,
  - 1 представляет свежий апельсин,
  - 2 представляет протухший апельсин.
- Каждую минуту свежий апельсин, соседствующий (по 4-ём направлениям) с протухшим, становится тоже протухшим.
- Вернуть минимальное количество минут, через которое все свежие апельсины станут протухшими, или -1, если такое невозможно.

```
int orangesRotting(std::vector<std::vector<int>>& grid);
```

Ссылка:
https://interview.cups.online/live-coding/?room=87b28a54-f8cf-4c05-8edf-33322beef054

Максимальный элемент на каждом уровне (medium)

- Дан указатель на корень
- Найти максимальный элемент на каждом уровне

```
std::vector<int> largestValues(TreeNode* root);
```

Ссылка:

https://interview.cups.online/live-coding/?room=889838b1-d883-4811-b94f-29713315dc5c

Слияние k-списков (hard)

- Дан вектор k-списков, отсортированных по возрастанию
- Нужно создать новый список, отсортированных по возрастанию

```
ListNode* mergeKLists(std::vector<ListNode*>& lists);
```

Ссылка:

https://interview.cups.online/live-coding/?room=98638c32-9806-4ff5-8016-d9d9561ed6ff

Спасибо за внимание!

Вопросы?