

Homework 6

23 октября 2022 г.

Task 1

Обозначим очереди как a , b . Будем поддерживать следующий инвариант:

- После каждой операции b - пустая.
- Элементы в a находятся в том же порядке что мы их добавляли.

Тогда операции $\text{push}(x)$ и $\text{pop}(x)$ устроены следующим образом.

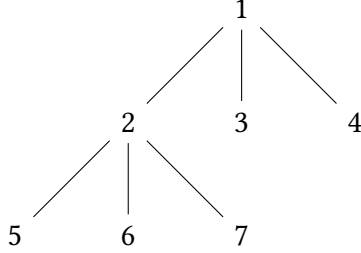
```
push(x) {  
    a.add(x)  
}
```

```
pop() {  
    while not a.empty() {  
        x = a.extract()  
        if not a.empty() {  
            b.add(x)  
        }  
    }  
    while not b.empty() {  
        a.add(b.extract())  
    }  
    return x  
}
```

То есть при добавлении элемента, мы добавляем его в конец a и инвариант сохраняется. При доставании элемента мы перекладываем все кроме последнего элемента в b и затем обратно в том же порядке в a , то есть инвариант сохраняется и возвращаем мы последний добавленный элемент.

Task 2

Аналогично пронумеруем все вершины. Докажем тогда, что вершина с номером x имеет детей с номерами $3x - 1, 3x, 3x + 1$. Соответственно родитель вершины x имеет номер $\lfloor \frac{x+1}{3} \rfloor$.



Пусть это верно для слоя h . Тогда самый средний лист этого слоя имеет индекс 3^{h-1} . Всего листов на этом слое 3^{h-1} , соответственно самый левый лист имеет индекс $l_h = 3^{h-1} - \frac{3^{h-1}-1}{2}$, самый правый лист имеет индекс $3^{h-1} + \frac{3^{h-1}-1}{2}$.

Тогда самый левый индекс слоя $h + 1$ имеет номер

$$\begin{aligned}
 3^{h-1} + \frac{3^{h-1} - 1}{2} + 1 &= \frac{2 * 3^{h-1} + 3^{h-1} + 1}{2} \\
 &= \frac{3^h + 1}{2} \\
 &= \frac{2 * 3^h - 3^h + 3 - 2}{2} \\
 &= 3^h - \frac{3^h - 3}{2} - 1 \\
 &= \left(3^{h-1} - \frac{3^{h-1} - 1}{2} \right) 3 - 1 \\
 &= l_h * 3 - 1
 \end{aligned}$$

Следовательно для самого левого листа слоя h и его левого ребенка формула работает.

Очевидно что для остальных его детей формула так же работает, то есть они имеют индексы $3l_h - 1, 3l_h, 3l_h + 1$. Тогда формула работает и для следующей вершины слоя h и тд, так как она имеет индекс $l_h + 1$, а ее дети $3l_h + 2, 3l_h + 3, 3l_h + 4 = 3(l_h + 1) - 1, 3(l_h + 1), 3(l_h + 1) + 1$.

База индукции видна на картинке.

Task 3

у - предок х

Пусть у не предок х. Тогда путь соединяющий х и у проходит через корень (обозначим корень г). Так как дерево бинарное, то они х и у по разным сторонам от г.

Таким образом

- либо $x < g < y$. Не может быть, так как у следующий по возрастанию от х.
- либо $x > g > y$. Не может быть, так как $x < y$.

Следовательно у - предок х.

у самый нижний, чей левый дочерний узел так же является предком х или самим х.

Так как у - предок х и $y > x$, то х находится в левом от у поддереве. Следовательно левый ребенок х является либо предком х либо сам является х.

Пусть z обладает тем же свойством но находится ниже у. Тогда т.к. у - предок х и z предок х, но ниже, то у - предок z и по доказанному ранее z находится в левом поддереве у. Тогда $x < z < y$, противоречье. Следовательно у самый низкий элемент обладающий таким свойством.

Task 4

a предок b либо b предок a

Рассмотрим вершину a . Докажем, что либо a предком b , либо b является предком a .

Аналогично предыдущей задаче, если они находятся по разные стороны от корня, то либо $a < r < b$, либо $a > r > b$, оба варианта противоречивы.

a предок b

Пусть a предок b . Так как $a < b$, то b находится в правом поддереве a . Обозначим левого и правого ребенка b как l и r . Тогда $a < l < b < r$. Противоречье.

b предок a

Пусть b предок a . Так как $a < b$, то a находится в левом поддереве b . Пусть у a есть правый ребенок r , тогда $a < r < b$. Противоречье, следовательно правого ребенка у нее нет.

Для c

Аналогично для c

- либо c предок b , либо b предок c
- c не может быть предком b
- у c не может быть левого ребенка

Task 5

1

Каждому множеству A необходимо сопоставить уникальную строку длины w . Пусть у разных множеств одинаковая строка. Тогда возьмем элемент который принадлежит одному множеству но не принадлежит другому. В обоих случаях мы получим одинаковый ответ от нашей структуры, соответственно один будет неверный.

Мощность множества подмножеств длины k равна $\binom{n}{k}$.

Мощность множества различных строк длины w равна 2^w .

Следовательно $2^w \geq \binom{n}{k}$.

Так же мы должны за t запросов определять лежит ли элемент в A .

Количество возможных пар (вопрос, ответ) длины t равно $2^{\binom{w}{t}}$.