Задача 8

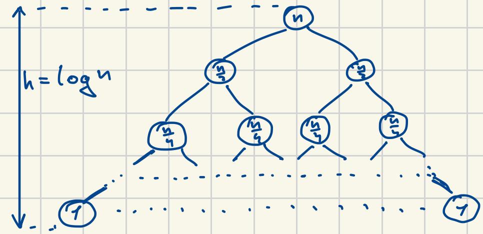
Задача трех T

Сложность некоторого рекурсивного алгоритма зависит от четности размера входных данных n и определяется следующим рекуррентным соотношением:

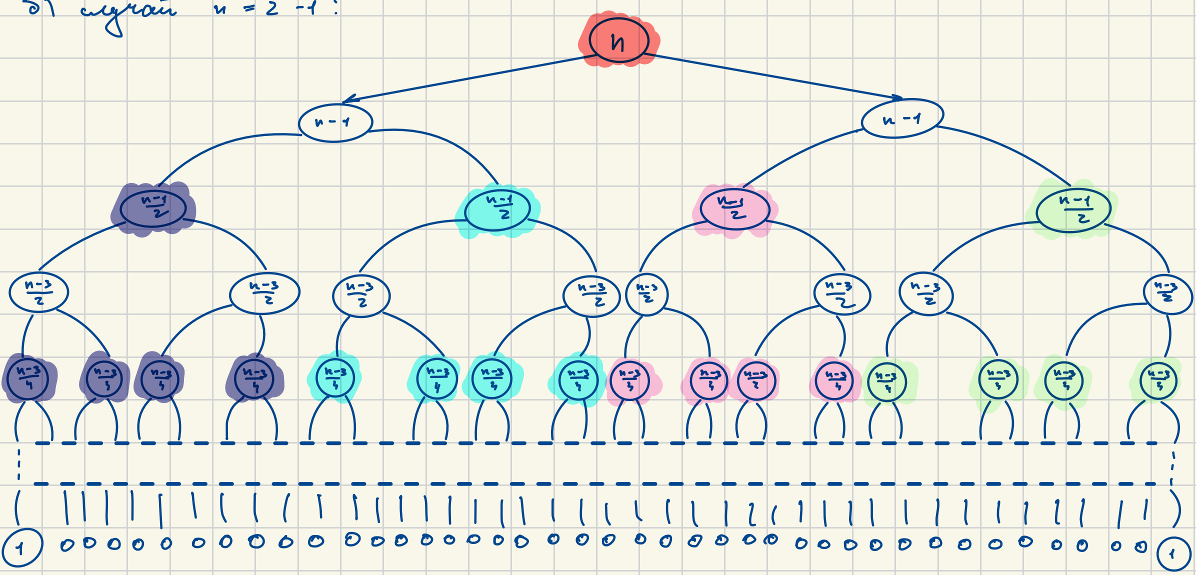
$$T(n) = \begin{cases} 2\,T(n/2) + n, \text{ если } n \text{ четно} \\ 2\,T(n-1) + n, \text{ если } n \text{ нечетно} \\ T(1) = 1 \end{cases}$$

1. Постройте дерево рекурсии для $T(n)$, учитывая четность n.
2. Обоснуйте верхнюю границу $O(g(n))$ для $T(n)$.

а) случай $n = 2^k$:



б) случай $n = 2^k - 1$:



2) Что бы док-ть верхнюю границу, необходимо разобрать случай $n = 2^k - 1$. В таком случае каждый раз будет $2T(n-1) + n$, а затем $2T(\frac{n}{2}) + n$.
Из пункта 1. и случая б) очевидно, что рекуррентную формулу для худшего случая можно переписать следующим образом: $T(n) = 4T(\frac{n-1}{2}) + n + 2\cdot(n-1) \Rightarrow$
$\Rightarrow T(n) = 4T(\frac{n-1}{2}) + 3n - 2$. Т.к. $T(n)$ – монотонна, то $T(\frac{n-1}{2}) \le T(\frac{n}{2}) \Rightarrow T(n) = 4T(\frac{n}{2}) + 2n^1$.
По мастер теореме: т.к. $\log_2 4 > 1$, то $T(n) = O(n^{\log_2 4}) \Rightarrow T(n) = O(n^2)$.

Ответ: $T(n) = O(n^2)$