

1 Кучи

- 1.1. Будем сливать обычные (двоичные) кучи следующим способом: при слиянии брать кучу меньшего размера и перекладывать ее элементы во вторую кучу. Сколько времени будут работать n таких операций?
- 1.2. Придумайте, как добавить в двоичную кучу k элементов за время $O(k + \log n)$. Как это поможет в предыдущей задаче?
- 1.3. Сколько вершин ранга r в биномиальном дереве ранга k ?
- 1.4. Разработайте алгоритм слияния биномиальных куч, который идет от больших деревьев к меньшим, с тем же временем работы.
- 1.5. В биномиальную кучу размера n добавляется один элемент. Как быстро понять, сколько операций слияния биномиальных деревьев произойдет?
- 1.6. Пусть подряд выполняется n операций **insert** в пустую биномиальную кучу. Какое среднее время операции?
- 1.7. Докажите, что амортизированная стоимость операции **insert** равна $O(1)$ (при том, что амортизированная стоимость остальных операций $O(\log n)$).
- 1.8. Как можно модифицировать биномиальную кучу, чтобы **insert** выполнялось за истинное $O(1)$, а амортизированная стоимость остальных операций не менялась?
- 1.9. Покажите, как сделать сливаемую кучу на базе любого сливаемого сбалансированного дерева поиска (например, декартового, или 2-3 дерева) с временем работы всех операций $O(\log n)$.
- 1.10. **NEW!** Докажите, что если $N_{-1} = N_0 = 1$, $N_k = \sum_{i=-1}^{k-2} N_i$, то $N_k = \text{Fib}_{k+1}$.
- 1.11. **NEW!** Покажите, что существует последовательность операций, приводящая к появлению в фибоначчиевой куче дерева в виде цепочки из n элементов (бамбука).
- 1.12. **NEW!** Разрешим удалять у вершины не одного ребенка, а k детей (k — константа). Изменится ли асимптотика времени работы фибоначчиевой кучи?
- 1.13. **NEW!** Добавьте в фибоначчиеву кучу операцию **increaseKey**, увеличивающую ключ элемента. Какого минимального времени работы вы умеете добиваться?
- 1.14. **NEW!** Тонкие кучи. Будем называть дерево «тонким», если оно может быть получено из биномиального удалением у некоторых вершин ребенка максимального ранга. Тонкой кучей называется коллекция тонких деревьев. Ограничений на число деревьев одного ранга нет. Разработайте операции **merge** и **extractMin** для тонких куч. Амортизированная стоимость операции **extractMin** должна быть $O(\log n)$. Амортизированная стоимость операции **merge** должна быть $O(1)$.
- 1.15. **NEW!** Разработайте операцию **decreaseKey** для тонкой кучи. Докажите, что амортизированное время выполнения есть $O(1)$ (используйте потенциал $2M + T$, где M — число вершин, у которых удалили ребенка)
- 1.16. **NEW!** Докажите, что операция **decreaseKey** в тонкой куче из предыдущего задания выполняется за истинные $O(\log n)$