Задача 1

Нам дали кучу A
Переопределим операцию extract_min для A (про остальные операции забудем, потому что не нужны)
При извлечении максимального элемента, если куча одна, то разделим её на две кучи и вернём значение в старом корне.
Пусть у нас есть i куч, оставшиеся от A после $i-1$ операций extract_min.
Что-бы извлечь минимум, нужно найти кучу с минимальным элементом и проделать операцию как в случае с одной кучей.
Нужно искать кучу с минимальным корнем за "быстро". Заведём ещё одну кучу, в которую будем добавлять новые корни и извлекать минимальный корень.
Что-бы найти k-й в исходной куче, нужно сделать k новых extract_min.
Итого на каждом extract_min у нас во вспомогательной куче лежит не более k элементов $\Rightarrow$ извлекаем минимум и добавляем вершины за $\mathcal{O}(\log_2 k)$
Так как у нас k операций extract_min, то алгоритм работает за $\mathcal{O}(k \log_2 k)$

Задача 2 (тут немного, но это честная работа)

Будем хранить в вершине новое поле — сколько нужно добавить к этому поддереву.
В нужный момент будем проталкивать эту информацию детям.
Когда нужно добавить в поддерево v число d — прибавляем d в поле add вершинки x
Пусть у нас для кучи будет храниться сколько нужно прибавить
Как-нибудь надо пушить значения (типо ДО), но получается дичь с decreaseKey, по этому не решил.

Задача 3

Создадим структуру из двух куч одинакового размера (при нечетном суммарном количестве — в левой больше элементов).
Правая — минимальная, левая — максимальная.
Инвариант: в правой куче все элементы больше чем в первой.
Из инварианта и структуры следует что в корне левой кучи всегда будет находиться искомая медиана.

Добавляем

- При добавлении элемента x будем смотреть как он соотносится с корнем левой кучи l.
  $x > l, \Rightarrow$ добавим x в правую кучу.
  $x \leq l, \Rightarrow$ добавим x в левую кучу.

- Теперь нарушился инвариант на соотношение количеств элементов в ку-

чах. Ну давайте просто из той, в которой много извлечем max(min) элемент из той кучи, в которой слишком много элементов и перекиенм в ту, в которой их не хватает.
Так как баланс изменился не более чем н 1 (мы добавили всего один элемент), то и перекинем таким образом мы не более одного элемента.

Удаляем медиану
Сделаем extract_min из левой кучи и как при добавлении перекинем нужный элемент из большей кучи в меньшую.

Возвращаем медиану
Ну теперь совсем изи. По следствию у нас в корне левой кучи ровно медиана.

Все запросы работают за $\mathcal{O}(\log_2 n)$

Задача 4 (тут тоже не много, но это всё что есть)
Если матрица не изменяется, то:

- Нечетные строки отсортированы по возрастанию
- Четные строки отсортированы по убыванию
- Каждый столбец отсортирован по возрастанию

Ку-ка-ре-ку, видим что матрица будет ввиде змейки.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 9 & 8 & 7 & 6 & 5 \\ 10 & 11 & 12 & 13 & 14 \\ 19 & 18 & 17 & 16 & 15 \end{bmatrix}$$

Это *нетрудно* заметить, если посмотреть различные случаи.