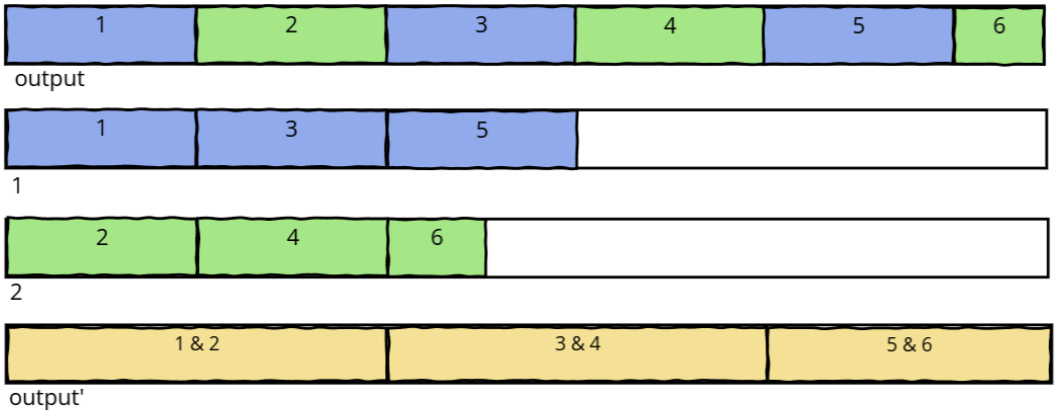
Сортировка слиянием

Сортировка слиянием реализована без рекурсии с двумя дополнительными лентами. Входную и выходную ленты назовем, соответственно, `input` и `output`, две дополнительные ленты будем называть `extra1` и `extra2` (на рисунках однако будем обозначать их просто цифрами).



Алгоритм разделяется на несколько шагов.

1. **In-memory сортировка.** Разделяем входные данные на блоки длины M. Последовательно загружаем блоки в оперативную память и сортируем их. После записываем последовательно на выходную ленту.
2. **Слияния.** Данный шаг мы повторяем, пока количество блоков > 1. В начале «слияния» на выходной ленте записано несколько блоков длины $2^k M$. Чередуя копируем блоки с выходной ленты на дополнительные (`extra1`, `extra2`). После чего последовательно сливаем соответствующие блоки на выходную ленту. Получаем конфигурацию выходной ленты с блоками размеров $2^{k+1} M$ (количество блоков уменьшилось).

Оценим количественную сложность алгоритма.

Шаг 1 выполняется за $O_{\text{RAM}}(N \log(M)) + O_{\text{read}}(N) + O_{\text{write}}(N) + O_{\text{move}}(N)$, где под $O(f)$ подразумевается сложность на устройстве с оперативной памятью, а $O_{\text{something}}(f)$ – сложность соответствующей операции на ленточном хранилище (если быть до конца честным, то запись выше означает сокращение слов «... в оперативной памяти работает за $O(N \log(M))$, делает $O(N)$ чтений, делает $O(N)$ записей, $O(N)$ раз двигает головку»).

Шаг 2, очевидно, выполняется за $O_{\text{RAM}}(N) + O_{\text{read}}(N) + O_{\text{write}}(N) + O_{\text{move}}(N)$. Всего же таких шагов будет сделано $O\left(\log\left(\frac{N}{M}\right)\right)$.

Итоговая сложность алгоритма тогда $O_{\text{RAM}}(N \log(N)) + O_{\text{read}}\left(N \log\left(\frac{N}{M}\right)\right) + O_{\text{write}}\left(N \log\left(\frac{N}{M}\right)\right) + O_{\text{move}}\left(N \log\left(\frac{N}{M}\right)\right)$. То есть количество доступной оперативной памяти сказывается на количестве операций, производимых с ленточным хранилищем.