число вытягиваний и шагов) сортирует состав, в котором после перенумерации n блоков. И наоборот. (Под «так же» понимается, что на каждом шаге вагоны с тем же самым номером отправляются в тот же самый стек и вытягиваются вагоны из тех же стеков в том же порядке.)

Очевидно, что нужно наблюдать за крайним правым вагоном из i-го блока и крайним левым вагоном из $(i+1)$-го. Рассуждения такие же, как и в лемме 2.

Из леммы 3 мы получаем уже совершенно нетривиальное

Следствие 1. *Общую задачу решает алгоритм, оптимально сортирующий состав, вагоны в котором расположены в обратном порядке.*

Итак, доказана

Теорема 2. *Любую перестановку σ можно отсортировать на горке с k стеками за не более чем s шагов, где $k^{s-1} < n \le k^s$, а n — число блоков, и эта оценка неулучшаема.*

Итак, задача об алгоритме сортировки за наименьшее число шагов решена, причем не только в целом, но и для каждой конкретной перестановки.

Замечание 2. Кто из вас, дорогой читатель, не играл хоть раз в жизни в карты? Наверное, все когда-нибудь держали в руках колоду карт и тасовали ее. Существует несколько способов тасования колоды карт. Один из них называется врезка. Колоду карт делят на две, не обязательно равные, части. После этого одну из частей, не меняя в ней порядок, вставляют в другую так, что карты из разных частей перемежаются.

А задавал ли кто-нибудь из вас себе такой вопрос: за какое наименьшее число врезок колоду можно перевести из одного положения в другое?

Я думаю, вы уже догадались, что ответ и на этот вопрос содержится в этой статье. Ведь операция врезки — это операция обратная роспуску и последующему вытягиванию из обоих стеков на горке с двумя стеками.

Задача 10. Сформулируйте ответ на поставленный вопрос.

А как же наименьшее число вытягиваний? Сортируя состав из n блоков по алгоритму поразрядной сортировки на горке с k стеками, мы делаем $\left[\log_k n\right]$ шагов и $k\times\left[\log_k n\right]$ вытягиваний. Это число равно $k\times\left[\frac{1}{\ln k}\ln n\right]$ (вот зачем нужно уметь пользоваться свойствами логарифмов). Конечно, число вытягиваний — целое число, но мы рассмотрим функцию $\frac{k}{\ln k}\ln n$. Последний сомножитель $\ln n$ не зависит от k, поэтому посмотрим на коэффициент перед $\ln n$: $g(k) = \frac{k}{\ln k}$. Более того, рассмотрим эту функцию не только при целых k, но и при любых действительных x.

Задача 11. Докажите, что функция $g(x) = \frac{x}{\ln x}$ имеет минимум при $x = e$.

Ближайшие к e целые числа — это 2 и 3. При этом $\frac{2}{\ln 2} \ge \frac{3}{\ln 3}$ (это упражнение). Поэтому для алгоритма поразрядной сортировки самое выгодное k в смысле вытягиваний — это $k = 3$.

Но так уж устроен человек — ему мало знать, что при больших n он может отсортировать состав примерно за $\frac{3}{\ln 3}\ln n$ вытягиваний. Хочется знать: а каково наименьшее число вытягиваний p для данных конкретных n и k?

Вспомним, что леммы 1, 2 и 3 были применимы не только к шагам сортировки, но и к вытягиваниям. Также для вытягиваний верно и следствие 1. Поэтому будем в дальнейшем предполагать, что в начале любой сортировки вагоны нашего состава расположены в обратном порядке.

Попробуем обратить наш вопрос: а каково наибольшее число вагонов, идущих в обратном порядке, которые можно отсортировать при данных p и k?

Итак, введем еще одно обозначение.

Определение 3. Обозначим через $W(p, k)$ максимальное число вагонов n, идущих в последовательности n, $n-1$, ..., 2, 1, которые можно отсортировать за p вытягиваний, имея k стеков.

Вот некоторые очевидные равенства для $W(p, k)$:

$$W(p, 1) = 1,\ W(1, k) = 1\ ,$$
$$W(0, k) = 1.$$

Первое равенство означает, что на одном стеке можно отсортировать только 1 вагон, второе — что за одно вытягивание можно отсортировать тоже только 1 вагон, и третье — что за 0 шагов можно отсортировать тоже только 1 вагон.

Лемма 4. *$W(2, 2) = 2$.*

Меньшим числом не обойтись, а за 2 вытягивания ясно как сделать.

Лемма 5. *Если $p < k$, то $W(p, k) = W(p, p)$.*

Действительно, если мы можем использовать p вытягиваний, то и стеков можем использовать не больше p, ведь из каждого использованного стека вагоны когда-нибудь придется вытягивать.

Будем в процессе решения нашей задачи составлять таблицу, где в p-й строке и k-м столбце стоит $W(p, k)$.

Следующее утверждение является основанием для дальнейшей индукции.

Теорема 3.

$$W(k+1,\ k+1) = 2W(k,\ k).$$

Будем доказывать это утверждение по индукции.

1°. Для $k = 1$ это верно: $2 = W(2,2) = 2W(1,1) = 2\times 1$.

2°. Предположим, мы умеем сортировать состав, идущий в обратном порядке, из $W(k, k)$ вагонов на горке из k стеков за k вытягиваний. Назовем этот алгоритм $(A(k))$. Рассмотрим теперь на горке с $(k+1)$ стеком состав из $2\times W(k, k)$ вагонов. Напишем номера всех вагонов в двоичной системе, начиная с нуля (напомним, что нулевой вагон самый правый). Тогда алгоритм $(A(k+1))$ состоит в следующем:

(1) Все четные вагоны в первом роспуске помещаются в $(k+1)$-й стек.

(2) У нечетных вагонов отбрасывается последний разряд (т.е. номер $2m+1$ переходит в номер m), и они распускаются в стеки с 1-го по k-й в соответствии с алгоритмом $(A(k))$.

(3) Вагоны из $(k+1)$-го стека вытягиваются, у них отбрасывается последний разряд (т.е. номер $2m$ переходит в номер m), и они распускаются в стеки с 1-го по k-й в соответствии с алгоритмом $(A(k))$.

(4) Дальше вагоны сортируются в соответствии с алгоритмом $(A(k))$:

Предыдущее рассуждение доказывает, что $W(k+1,\ k+1) \ge 2\times W(k,\ k)$.

3°. Докажем теперь обратное неравенство: $W(k+1,\ k+1) \le 2\times W(k,\ k)$.

Действительно, пусть на горке стоит состав из $n > 2\times W(k, k)$ вагонов. Тогда после первого роспуска и первого вытягивания либо на горке состав из $n_1 > W(k, k)$ вагонов, либо в