оставшихся k стеках находится суммарно $n_2 > W(k, k)$. В каждом из этих случаев за оставшиеся k вытягиваний не удастся отсортировать эти вагоны. Теорема 3 доказана.

Следствие 2. $W(k, k) = 2^{k-1}$.

Мы уже можем заполнить большую часть нашей таблицы. Вот как теперь она выглядит (рис.7).

Число вытягиваний (p) \ Число стеков (k)	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	2	2	2	2	2	2	2
3	1		4	4	4	4	4	4
4	1			8	8	8	8	8
5	1				16	16	16	16
6	1					32	32	32
7	1						64	64

Рис. 7

Попробуем применить идею предыдущего доказательства к случаю горки с двумя стеками.

Теорема 4. *$k = 2$. Пусть $F(p)$ — p-е число Фибоначчи, т.е. p-е число из последовательности $F(p)$, удовлетворяющей рекуррентному соотношению $F(p+1) = F(p) + F(p-1)$ и начальным данным $F(0)=1$, $F(1) = 1$, $F(2) = 2$. Тогда $W(p, 2) = F(p)$.*

Будем доказывать и это утверждение по индукции.

1°. $W(1, 2) = 1 = F(1)$, $W(2, 2) = 2 = F(2)$. Можно еще проверить, что $W(3, 2) = 3$.

Пусть теперь при всех $2 \le l < p$ за l вытягиваний можно отсортировать состав из не более $F(l)$ вагонов, идущих в обратном порядке. Докажем, что то же верно и для $l = p$.

2°. Докажем, что $W(p, 2) \le F(p)$, т.е. состав из большего числа вагонов отсортировать нельзя. Пусть это не так. Пусть $n > F(p)$. Тогда после первого роспуска в стеках оказалось $n_1 \ge n_2$ вагонов. Так как $n > F(p)$, а $F(p) = F(p-1) + F(p-2)$, то хотя бы одно из чисел n_i строго больше, чем соответствующее число $F(p-i)$, $i = 1,2$.

а) Если $n_1 > F(p-1)$, то после первого же вытягивания у нас есть «состав» из n_1 вагонов, идущих в обратном порядке, который по предположению индукции нельзя отсортировать за $p-1$ вытягивание.

б) Если $n_2 > F(p-2)$, то после второго вытягивания у нас на горке окажется неотсортированный состав из не менее n_2 вагонов, идущих в обратном порядке, который мы не можем по предположению индукции отсортировать за $p-2$ вытягивания.

Это же рассуждение позволяет рекурсивно построить алгоритм, с помощью которого за $p+1$ вытягивание можно отсортировать состав из $F(p+1)$ вагонов, идущих в обратом порядке.

3°. Пусть алгоритм $(B(p))$ сортирует состав из $F(p)$ вагонов, идущих в обратном порядке. Пусть после первого роспуска алгоритма $(B(p))$ в первом стеке окажется $F(p-1)$ вагон, а во втором — $F(p-2)$. Покрасим вагоны в первом стеке в синий цвет, а во втором — в желтый, и вернем ситуацию назад во времени. У нас на горке стоит $F(p)$ вагонов. Вставим после каждого синего вагона зеленый (т.е. зеленый вагон правее синего), и перенумеруем состав в убывающем порядке. Теперь у нас на горке стоит состав из $F(p+1) = F(p) + F(p-1)$ вагонов.

Алгоритм $(B(p+1))$ будет работать так:

(1) во время первого роспуска синие вагоны отправляются в 1-й стек, а желтые и зеленые во второй;

(2) вагоны из второго стека вытягиваются (1-е вытягивание);

(3) зеленые вагоны распускаются в первый стек, а желтые во второй.

Теперь в 1-м стеке все зеленые вагоны стоят левее синих.

Перенумеровав вагоны в 1-м и 2-м стеках, мы получим ситуацию после первого роспуска в алгоритме $(B(p))$. Дальше сортируем с помощью алгоритма $(B(p))$.

Теорема 4 доказана.

Теперь уже ничего не стоит сформулировать аналог теоремы 4 для общего случая. Как вы уже догадались, общий алгоритм является обобщением сортировки Фибоначчи.

Определение 4. Введем последовательность $u_k(k) = 2^{k-1}$, задав ее рекуррентно:

$$u_k(p) = u_k(p-1) + u_k(p-2) + \ldots + u_k(p-k), \quad p > k,$$
$$u_k(1) = 1, \ \ldots, \ u_k(k) = 2^{k-1}. \qquad (*)$$

Теорема 5. $W(p, k) = u_k(p)$.

Доказательство этого утверждения естественно обобщает доказательство теоремы 4. Его мы тоже проведем по индукции.

1°. При $p \le k$ наше утверждение следует из теоремы 3, леммы 5, следствия 2 и равенства $(*)$.

Рассмотрим случай $p > k$. Пусть при всех $k < l < p$ за l вытягиваний можно отсортировать состав из не более $u_k(l)$ вагонов, идущих в обратном порядке. Докажем, что то же верно и для $l = p$.

2°. Докажем, что $W(p, k) \le u_k(p)$, т.е. состав из большего числа вагонов отсортировать нельзя. Пусть это не так. Пусть $n > u_k(p)$ и пусть после первого роспуска в стеках оказалось $n_1 \ge n_2 \ge n_3 \ge \ldots \ge n_k$ вагонов. Сравним эту последовательность с последовательностью $u_k(p-1) \ge u_k(p-2) \ge \ldots \ge u_k(p-k)$. Так как $n > u_k(p)$, то существует номер m такой, что $n_m > u_k(p-m)$, а это означает, что среди чисел $n_1 \ge n_2 \ge n_3 \ge \ldots \ge n_k$ не меньше m строго больших $u_k(p-m)$. Следовательно, после m вытягиваний мы будем иметь или на горке, или в одном из стеков последовательность из не менее $u_k(p-m) + 1$ вагона, номера которых идут в убывающем порядке, и которую, по индукционному предположению, мы не можем отсортировать за оставшиеся $p - m$ вытягиваний.

Так же, как и в теореме 4, строится рекурсивный алгоритм.

3°. Пусть алгоритм $(C_k(p))$, $p > k$, сортирует состав из $u_k(p)$ вагонов, идущих в обратном порядке. Пусть после первого роспуска алгоритма $(C_k(p))$ в первом стеке окажется $u_k(p-1)$ вагон, во втором — $u_k(p-2)$ и т.д., в k-м — $u_k(p-k)$. Покрасим вагоны в первых $(k-1)$ стеках в синий цвет, а в последнем — в желтый, и вернем ситуацию назад во времени. У нас на горке стоит $u_k(p)$ вагонов. Вставим после каждого синего вагона зеленый (т.е. зеленый вагон правее синего), перенумеруем состав в убывающем порядке. Теперь у нас на горке стоит состав из

$$u_k(p+1) = u_k(p) + \left(u_k(p-1) + u_k(p-2) + \ldots + u_k(p-k+1)\right)$$

вагонов.

Алгоритм $(C_k(p+1))$ работает так:

(1) во время первого роспуска синие вагоны отправляются в первые $(k-1)$ стеки в соответствии с алгоритмом $(C_k(p))$, а желтые и зеленые в последний;

(2) вагоны из последнего стека вытягиваются (1-е вытягивание);