B13 (повышенный уровень, время –7 мин)

Тема: динамическое программирование.

Что нужно знать:

- динамическое программирование – это способ решения сложных задач путем сведения их к более простым задачам того же типа
- с помощью динамического программирования решаются задачи, которые требуют полного перебор вариантов:
  - «подсчитайте количество вариантов…»
  - «как оптимально распределить…»
  - «найдите оптимальный маршрут…»
- динамическое программирование позволяет ускорить выполнение программы за счет использования дополнительной памяти; полный перебор не требуется, поскольку запоминаются решения всех задач с меньшими значениями параметров

Пример задания:

У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1
2. умножь на 3

Первая из них увеличивает число на экране на 1, вторая – утраивает его.
Программа для Утроителя – это последовательность команд.
Сколько есть программ, которые число 1 преобразуют в число 20?

Решение (1 способ, составление таблицы):

1) заметим, что при выполнении любой из команд число увеличивается (не может уменьшаться)
2) начнем с простых случаев, с которых будем начинать вычисления: для чисел 1 и 2, меньших, чем 3, существует только одна программа, состоящая только из команд сложения; если через K_N обозначить количество разных программ для получения числа *N* из 1, то $K_1 = K_2 = 1$.
3) теперь рассмотрим общий случай, чтобы построить рекуррентную формулу, связывающую K_N с предыдущими элементами последовательности $K_1, K_2, \ldots, K_N$, то есть с решениями таких же задач для меньших N
4) если число N не делится на 3, то оно могло быть получено только последней операцией сложения, поэтому $K_N = K_{N-1}$
5) если N делится на 3, то последней командой может быть как сложение, так и умножение
6) поэтому для получения K_N нужно сложить K_{N-1} (количество программ с последней командой сложения) и $K_{N/3}$ (количество программ с последней командой умножения). В итоге получаем:
   если N не делится на 3: $K_N = K_{N-1}$
   если N делится на 3: $K_N = K_{N-1} + K_{N/3}$
7) остается заполнить таблицу для всех значений от 1 до N:

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
K_N	1	1	2	2	2	3	3	3	5	5	5	7	7	7	9	9	9	12	12	12

8) Заметим, что количество вариантов меняется только в тех столбцах, где N делится на 3, поэтому из всей таблицы можно оставить только эти столбцы:

N	1	3	6	9	12	15	18	21
K_N	1	2	3	5	7	9	12	15

9) заданное число 20 попадает в последний интервал (от 18 до 21), поэтому …
10) ответ – 12.

Решение (2 способ, подстановка – вычисления по формулам «с конца»):

1) п. 1-6 выполняются так же, как и при первом способе; главная задача – получить рекуррентную формулу:
   если N не делится на 3: $K_N = K_{N-1}$
   если N делится на 3: $K_N = K_{N-1} + K_{N/3}$
   с начальными условиями $K_1 = K_2 = 1$
2) начинаем с заданного конечного числа 20; применяем первую формулу ($K_N = K_{N-1}$), пока не дойдем до числа, делящегося на 3 (это 18):
   $$K_{20} = K_{19} = K_{18}$$
3) далее применяем вторую формулу ($K_N = K_{N-1} + K_{N/3}$):
   $$K_{20} = K_{18} = K_{17} + K_6$$
4) применяем первую формулу для 17:
   $$K_{17} = K_{16} = K_{15} \quad \Rightarrow \quad K_{20} = K_{15} + K_6$$
5) применяем вторую формулу для обоих слагаемых:
   $$K_{20} = (K_{14} + K_5) + (K_5 + K_2) = K_{14} + 2K_5 + 1$$
   где учтено, что $K_2 = 1$
6) с помощью первой формулы переходим в правой части к числам, делящимся на 3:
   $$K_{20} = K_{12} + 2K_3 + 1$$
   а затем применяем вторую формулу для каждого слагаемого
   $$K_{20} = (K_{11} + K_4) + 2(K_2 + K_1) + 1 = K_{11} + K_4 + 2(1+1) + 1 = K_{11} + K_4 + 5$$
7) снова используем первую формулу
   $$K_{20} = K_9 + K_3 + 5$$
   а затем – вторую:
   $$K_{20} = (K_8 + K_3) + (K_2 + K_1) + 5 = K_8 + 2(K_2 + K_1) + 5 = K_8 + 9$$
8) и еще раз
   $$K_{20} = K_6 + 9 = K_5 + K_2 + 9 = K_5 + 10 = K_3 + 10 = 2 + 10 = 12$$
9) ответ – 12.

Решение (3 способ, О.В. Шецова, лицей № 6, г. Дубна):

1) будем составлять таблицу из трех столбцов: в первом записывается получаемое число от 1 до 20, во втором – какой последней командой может быть получено это число, а в третьем вычисляем количество различных программ для получения этого числа из 1
2) очевидно, что число 1 может быть получено с помощью одной единственной (пустой) программы:

Число	Как можно получить?	Количество программ
1		1

3) число 2 не делится на 3, поэтому его можно получить только командой сложения (+1), значит, количество программ для 2 совпадает с количеством программ для 1:

Число	Как можно получить?	Количество программ
1		1
2	+1	= 1

4) число 3 делится на 3, поэтому его можно получить с помощью двух команд: +1 (из 2) и *3 (из 1):

Число	Как можно получить?	Количество программ
1		1
2	+1	1
3	+1 *3	1 + 1 = 2

5) числа 4 и 5 не делятся на 3, поэтому их можно получить только с помощью команды +1, а число 6 может быть получено двумя командами:

Число	Как можно получить?	Количество программ
1		1
2	+1	1
3	+1 *3	1 + 1 = 2
4	+1	2
5	+1	2
6	+1 *3	2 + 1 = 3

6) следующая группа – 7, 8 (не делятся на 3) и 9 (делится на 3):

Число	Как можно получить?	Количество программ
1		1
2	+1	1
3	+1 *3	1 + 1 = 2
4	+1	2
5	+1	2
6	+1 *3	2 + 1 = 3
7	+1	3
8	+1	3
9	+1 *3	3 + 2 = 5

7) далее – 10, 11 и 12:

Число	Как можно получить?	Количество программ
1		1
2	+1	1
3	+1 *3	1 + 1 = 2
4	+1	2
5	+1	2
6	+1 *3	2 + 1 = 3
7	+1	3
8	+1	3
9	+1 *3	3 + 2 = 5
10	+1	5
11	+1	5
12	+1 *3	5 + 2 = 7

8) и так далее, вот полностью заполненная таблица (до конечного числа 20):

Число	Как можно получить?	Количество программ
1		1
2	+1	1
3	+1 *3	1 + 1 = 2
4	+1	2
5	+1	2
6	+1 *3	2 + 1 = 3
7	+1	3
8	+1	3
9	+1 *3	3 + 2 = 5
10	+1	5
11	+1	5
12	+1 *3	5 + 2 = 7
13	+1	7
14	+1	7
15	+1 *3	7 + 2 = 9
16	+1	9
17	+1	9
18	+1 *3	9 + 3 = 12
19	+1	12
20	+1	12

9) ответ – количество программ, с помощью которых можно получить число 20 из 1, – считываем из последней ячейки третьего столбца
10) ответ – 12.

Решение (4 способ, М.В. Кузнецова и её ученики, г. Новокузнецк):

1) пусть N – искомое конечное число, $K(N)$ количества программ получения числа N
2) тогда для построения рекуррентной формулы определения $K(N)$, нужно знать 2 факта:
   а) какой может быть последняя команда и сколько есть видов этого последнего действия?
   б) для каждого «последнего» действия нужно знать число программ получения предыдущего числа, сумма этих количеств и есть искомое значение $K(N)$ – число программ получения числа N.

   Например, общее количество программ получения числа 6 с помощью Утроителя равно $K(6) = K(5) + K(2)$, т.к. есть ДВА способа завершения программ получения этого значения: 6=5+1 и 6=2·3 .
3) число программ получения числа N зависит от числа программ получения предыдущего значения, и что программы получения чисел, кратных 3-м могут завершаться 2-мя способами: $(N-1)+1$ или $(N/3)\cdot 3$, а все остальные числа получают только первым способом: $(N-1)+1$.
4) составим рекуррентную формулу для определения числа программ получения числа N:
   при $N = 1$ имеем $K(1) = 1$
   если N не кратно 3: $K(N) = K(N-1)$
   если N делится на 3: $K(N) = K(N-1) + K(N/3)$
5) с помощью это формулы заполняем таблицу следующим образом:
   – в первом столбце записываем все натуральные числа от 1 до заданного N;
   – во втором столбце – числа, на единицу меньшие (из которых может быть получено N последней операцией сложения с 1);
   – в третьем столбце для чисел, кратных 3-м, записываем частное от деления числа, записанного в первом столбце, на 3 (из этого числа может быть получено N последней операцией умножения на 3);
   – в последнем столбце вычисляем $K(N)$, складывая соответствующие значения для тех строк, номера которых записаны во втором и третьем столбцах:

N	**N-1**	**N/3**	**K(N)**
1	–		**1**
2	1		**1**
3	2	1	1+1=**2**
4	3		**2**
5	4		**2**
6	5	2	2+1=**3**
7	6		**3**
8	7		**3**
9	8	3	3 + 2=**5**
10	9		**5**
11	10		**5**
12	11	4	5 + 2 = **7**
13	12		**7**
14	13		**7**
15	14	5	7 + 2 = **9**
16	15		**9**
17	16		**9**

18	17	6	9+3 = **12**
19	18		**12**
20	19		**12**

6) ответ – 12.

Решение (5 способ, А. Сидоров):

1) основная идея – число программ, преобразующих начальное число 1 в конечное 20 с помощью заданных в условии команд, равно числу программ, преобразующих конечное число 20 в начальное 1 с помощью обратных команд: «**`вычти 1`**» и «**`раздели на 3`**»
2) будем строить «обратное дерево» – дерево всех способов преобразования **конечного числа в начальное**; это лучше (в сравнении с построением «прямого» дерева, от начального числа к конечному), потому что операция умножения необратима – каждое число можно умножить на 3, но не каждое можно разделить на 3; из-за этого сразу отбрасываются тупиковые ветви, не дающие новых решений
3) рисуем сокращенное дерево, в котором черные стрелки показывают действие первой команды («прибавь 1»), а красные – действие второй команды («умножь на 3»); красные стрелки подходят только к тем числам, которые делятся на 3:

```
20←19←18←17←16←15←14←13←12←11←10←9←8←7←6←5←4←3←2←1
      ↑          ↑          ↑         ↑     ↑     ↑
      6          5          4         3     2     1
```

4) чтобы получить количество программ для каждого числа из верхней строки, нужно сложить соответствующие количества программ для всех чисел из нижнего ряда, которые не больше данного (программы с умножением), и добавить 1 (программа, состоящая из одних сложений)
5) очевидно, что для получения 1 существует одна (пустая) программа; тогда для числа 2 тоже получается одна программа, а для числа 3 – две программы:

```
(2) (1)
 3←2←1
 ↑
 1
(1)
```

6) далее, для чисел 4 и 5 получаем 2 программы (после числа 3 нет «разветвлений» – подходящих красных стрелок) , а для числа 6 – 3 программы, так как «подошло» еще одно разветвление (6 можно получить умножением 2 на 3), а для числа 2 мы уже подсчитали количество программ – оно равно 1:

```
(3)        (2) (1)
 6←5←4←3←2←1
 ↑         ↑
 2         1            1 + 1 + 1 = 3
(1)       (1)
```

7) находить число программ для следующих чисел нам уже не понадобится, потому что при умножении на 3 они дают числа, большие, чем заданное конечное число 20
8) запишем полученные результаты в самой нижней строке для всех множителей от 1 до 6:

```
                                                 (3)    (2) (1)
20←19←18←17←16←15←14←13←12←11←10←9←8←7←6←5←4←3←2←1
      ↑          ↑          ↑         ↑     ↑     ↑
      6          5          4         3     2     1
     (3)        (2)        (2)       (2)   (1)   (1)
```

9) теперь остается сложить все числа в скобках в нижнем ряду (количество программ с командами умножения) и добавить 1 (одна программа, состоящая только из команд сложения):
   3 + 2 + 2 + 2 + 1 + 1 + 1 = 12
10) ответ – 12.

Возможные проблемы:
- неверно определенные начальные условия
- неверно выведенная рекуррентная формула
- ошибки при заполнении таблицы (невнимательность)
- второй способ (подстановка), как правило, приводит к бОльшему количеству вычислений; конечно, можно отдельно выписывать все полученные ранее значения K_N, но тогда мы фактически придем к табличному методу

Еще пример задания:

У исполнителя Калькулятор две команды, которым присвоены номера:

```
1. прибавь 1
2. увеличь вторую с конца цифру на 1
```

Первая из них увеличивает число на экране на 1, вторая – увеличивает на 1 число десятков.
Если перед выполнением команды 2 вторая с конца цифра равна 9, она не изменяется.
Программа для Калькулятора – это последовательность команд.
Сколько есть программ, которые число 15 преобразуют в число 28?

Решение (1 способ, составление таблицы):

1) заметим, что при выполнении любой из команд число увеличивается (не может уменьшаться)
2) при заданных командах очередное число N может быть получено двумя способами:
3) увеличением на 1 (для всех чисел, больших начального числа)
4) увеличением числа десятков на 1 (то есть, фактически командой «+10») – для всех чисел, больших или равных 25; например, число 24 не может быть получено этой командой (14 + 10 = 24), потому что число 14 меньше, чем начальное значение 15
5) таким образом, рекуррентные формулы принимают вид
   $K_N = K_{N-1}$ для всех чисел, меньших, чем 25
   $K_N = K_{N-1} + K_{N-10}$ для чисел, больших или равных 25
6) других способов получения числа с помощью исполнителя с заданными командами нет, то есть мы таким образом рассматриваем все возможные программы
7) начальное значение: $K_{15} = 1$ (число 15 можно получить единственной пустой программой)
8) далее заполняем таблицу:

N	15	16	17	18	19	20	21	22	23	24	25	26	27	28
K_N	1	1	1	1	1	1	1	1	1	1	2	3	4	5

9) Ответ: 5

Еще пример задания:

У исполнителя Калькулятор две команды, которым присвоены номера:

```
1. прибавь 1
2. увеличь две младшие цифры на 1
```

Первая из них увеличивает число на экране на 1, вторая – увеличивает на 1 число десятков и число единиц. Если перед выполнением команды 2 какая-либо из двух младших цифр равна 9, она не изменяется. Программа для Калькулятора – это последовательность команд.
Сколько есть программ, которые число 23 преобразуют в число 48?

Решение (1 способ, составление таблицы):

1) заметим, что при выполнении любой из команд число увеличивается (не может уменьшаться)
2) при заданных командах очередное число N может быть получено двумя способами:
3) увеличением на 1 (для всех чисел, больших начального числа)
4) увеличением обеих цифр на 1 в результате выполнения команды 2 (то есть, фактически командой «+11») – для всех чисел, больших или равных 23 + 11 = 34, которые **НЕ** оканчиваются на 0;
5) увеличением *только младшей* цифры на 1 в результате выполнения команды 2 (то есть, фактически командой «+1») – для всех чисел от 91 до 99, но в нашем диапазоне (23..48) таких нет
6) увеличением *только старшей* цифры на 1 в результате выполнения команды 2 (то есть, фактически командой «+10») – для всех чисел, больших 34 и имеющих 9 на конце; в нашем случае под этот вариант подходит только число 39
7) таким образом, рекуррентные формулы принимают вид

   $K_N = K_{N-1}$ для всех чисел, меньших, чем 34, а также для всех чисел, оканчивающихся на 0

   $K_N = K_{N-1} + K_{N-11}$ для чисел, больших или равных 34, кроме 39

   $K_N = K_{N-1} + K_{N-11} + K_{N-10}$ для числа 39
8) других способов получения числа с помощью исполнителя с заданными командами нет, то есть мы таким образом рассматриваем все возможные программы
9) начальное значение: $K_{23} = 1$ (число 23 можно получить единственной пустой программой)
10) далее заполняем таблицу:

N	23	…	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
K_N	1	…	1	2	3	4	5	6	8	8	9	10	11	12	14	17	21	26

    здесь многоточия означают, что для всех чисел от 23 до 33 включительно количество программ равно 1;
11) например, для числа 47 количество программ вычисляется как

    $K_{47} = K_{46} + K_{36}$ = 17 + 4 = 21

    а для числа 39 –как

    $K_{39} = K_{38} + K_{28} + K_{29}$ = 6 + 1 + 1 = 8
12) Ответ: 26