

### Задача 3 (Задача о куче).

Из камней весом  $p_i (i = 1, \dots, N)$  набрать кучу весом ровно  $W$  или, если это невозможно, максимально близкую к  $W$  (меньшую, чем  $W$ ).

Решение: Идея состоит в том, чтобы сохранять в массиве решения всех более простых задач этого типа (при меньшем количестве камней и меньшем весе  $W$ ).

Построим матрицу  $T$ , где элемент  $T[i,w]$  — это оптимальный вес, полученный при попытке собрать кучу весом  $w$  из  $i$  первых по счету камней. Очевидно, что первый столбец заполнен нулями (при заданном нулевом весе никаких камней не берем).

Рассмотрим строку (нет камней). Все значения строки - нули. Это простые варианты задачи, решения для которых легко подсчитать вручную.

Рассмотрим пример, когда требуется набрать вес 8 из камней весом 2, 4, 5 и 7

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0								
P2	4	0								
P3	5	0								
P4	7	0								

Теперь предположим, что строки с 1-й по  $(i-1)$ -ю уже заполнены. Перейдем к  $i$ -й строке, то есть добавим в набор  $i$ -й камень. Он может быть взят или не взят в кучу. Если мы не добавляем его в кучу, то  $T[i,w] = T[i-1,w]$ , то есть решение не меняется от добавления в набор нового камня. Если камень с весом  $p_i$  добавлен в кучу, то остается “добрать” остаток  $w-p_i$  оптимальным образом (используя только предыдущие камни), то есть  $T[i,w] = T[i-1,w-p_i] + p_i$ .

Как же выбрать, “брать или не брать”? Проверить, в каком случае полученное решение будет больше (ближе к  $w$ ). Таким образом, получается рекуррентная формула для заполнения таблицы:

при  $w < p_i$ :  $T[i,w] = T[i-1,w]$

при  $w \geq p_i$ :  $T[i,w] = \max(T[i-1,w], T[i-1,w-p_i] + p_i)$

Используя эту формулу, заполняем таблицу по строкам, сверху вниз; в каждой строке — слева направо:

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0	$T[1,1]$	$T[1,2]$	$T[1,3]$	$T[1,4]$	$T[1,5]$	$T[1,6]$	$T[1,7]$	$T[1,8]$
P2	4	0								
P3	5	0								
P4	7	0								

Имя Эл-га	Значение p	Условие	Формула вычисления	результат
$T[1,1]$	$W1=1$	$1 < 2$	$T[1,1]=T[0,1]$	0
$T[1,2]$	$W2=2$	$2=2$	$T[1,2]=\max(T[0,2], T[0,2-2]+2)=\max(0,0+2)$	2
$T[1,3]$	$W3=3$	$3 > 2$	$T[1,3]=\max(T[0,3], T[0,3-2]+2)=\max(0,0+2)$	2
$T[1,4]$	$W4=4$	$4 > 2$	$T[1,4]=\max(T[0,4], T[0,4-2]+2)=\max(0,0+2)$	2
$T[1,5]$	$W5=5$	$5 > 2$	$T[1,5]=\max(T[0,5], T[0,5-2]+2)=\max(0,0+2)$	2
$T[1,6]$	$W6=6$	$6 > 2$	$T[1,6]=\max(T[0,6], T[0,6-2]+2)=\max(0,0+2)$	2
$T[1,7]$	$W7=7$	$7 > 2$	$T[1,7]=\max(T[0,7], T[0,7-2]+2)=\max(0,0+2)$	2
$T[1,8]$	$W8=8$	$8 > 2$	$T[1,8]=\max(T[0,8], T[0,8-2]+2)=\max(0,0+2)$	2

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0	0	2	2	2	2	2	2	2
P2	4	0	T[2,1]	T[2,2]	T[2,3]	T[2,4]	T[2,5]	T[2,6]	T[2,7]	T[2,8]
P3	5	0								
P4	7	0								

Имя элемента		Значение p	Условие	Формула вычисления	результат
T[2,1]	W1=1	P2=4	1<4	T[2,1]=T[1,1]	0
T[2,2]	W2=2		2<4	T[2,2]=T[1,2]	2
T[2,3]	W3=3		3<4	T[2,3]=T[1,3]	2
T[2,4]	W4=4		4=4	T[2,4]=max(T[1,4],T[1,4-4]+4)=max(2,0+4)	4
T[2,5]	W5=5		5>4	T[2,5]=max(T[1,5],T[1,5-4]+4)=max(2,0+4)	4
T[2,6]	W6=6		6>4	T[2,6]=max(T[1,6],T[1,6-4]+4)=max(2,2+4)	6
T[2,7]	W7=7		7>4	T[2,7]=max(T[1,7],T[1,7-4]+4)=max(2,2+4)	6
T[2,8]	W8=8		8>4	T[2,8]=max(T[1,8],T[1,8-4]+4)=max(2,2+4)	6

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0	0	2	2	2	2	2	2	2
P2	4	0	0	2	2	4	4	6	6	6
P3	5	0	T[3,1]	T[3,2]	T[3,3]	T[3,4]	T[3,5]	T[3,6]	T[3,7]	T[3,8]
P4	7	0								

Имя элемента		Значение p	Условие	Формула вычисления	результат
T[3,1]	W1=1	P3=5	1<5	T[3,1]=T[2,1]	0
T[3,2]	W2=2		2<5	T[3,2]=T[2,2]	2
T[3,3]	W3=3		3<5	T[3,3]=T[2,3]	2
T[3,4]	W4=4		4<5	T[3,4]=T[2,4]	4
T[3,5]	W5=5		5=5	T[3,5]=max(T[2,5],T[2,5-5]+5)=max(4,0+5)	5
T[3,6]	W6=6		6>5	T[3,6]=max(T[2,6],T[2,6-5]+5)=max(6,0+5)	6
T[3,7]	W7=7		7>5	T[3,7]=max(T[2,7],T[2,7-5]+5)=max(6,2+5)	7
T[3,8]	W8=8		8>5	T[3,8]=max(T[2,8],T[2,8-5]+5)=max(6,2+5)	7

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0	0	2	2	2	2	2	2	2
P2	4	0	0	2	2	4	4	6	6	6
P3	5	0	0	2	2	4	5	6	7	7
P4	7	0	T[4,1]	T[4,2]	T[4,3]	T[4,4]	T[4,5]	T[4,6]	T[4,7]	T[4,8]

Имя элемента		Значение p	Условие	Формула вычисления	результат
T[4,1]	W1=1	P4=7	1<7	T[4,1]=T[3,1]	0
T[4,2]	W2=2		2<7	T[4,2]=T[3,2]	2

T[4,3]	W3=3		3<7	T[4,3]=T[3,3]	2
T[4,4]	W4=4		4<7	T[4,4]=T[3,4]	4
T[4,5]	W5=5		5<7	T[4,5]=T[3,5]	5
T[4,6]	W6=6		6<7	T[4,6]=T[3,6]	6
T[4,7]	W7=7		7=7	T[4,7]= max(T[3,7],T[3,7-7]+7)=max(7,0+7)	7
T[4,8]	W8=8		8>7	T[4,8]= max(T[3,8],T[3,8-7]+7)=max(7,0+7)	7

		W0	W1	W2	W3	W4	W5	W6	W7	W8
P0	0	0	0	0	0	0	0	0	0	0
P1	2	0	0	2	2	2	2	2	2	2
P2	4	0	0	2	2	4	4	6	6	6
P3	5	0	0	2	2	4	5	6	7	7
P4	7	0	0	2	2	4	5	6	7	7

Видим, что сумму 8 набрать невозможно, ближайшее значение — 7 (правый нижний угол таблицы).

Эта таблица содержит все необходимые данные для определения выбранной группы камней.

Действительно, если камень с весом  $pi$  не включен в набор, то  $T[i,w] = T[i-1,w]$ , то есть число в таблице не меняется при переходе на строку вверх. Начинаем с левого нижнего угла таблицы, идем вверх, пока значения в столбце равны 7. Последнее такое значение — для камня с весом 5, поэтому он и выбран. Вычитая его вес из суммы, получаем  $7 - 5 = 2$ , переходим во второй столбец на одну строку вверх и снова идем вверх по столбцу, пока значение не меняется (равно 2). Так как мы успешно дошли до самого верха таблицы, взят первый камень с весом 2.

```

program Project1;
CONST N=4;W=8;
VAR I,J,SUM:INTEGER;
P:ARRAY[0..N] OF BYTE;
T:ARRAY[0..N,0..W] OF WORD;
FUNCTION MAX(X,Y:WORD):WORD;
BEGIN
  IF X>Y THEN MAX:=X ELSE MAX:=Y;
END;
begin
P[0]:=0;
{OSTALN KAMNI - WWOD} P[1]:=2;P[2]:=4;P[3]:=5;P[4]:=7;
{-----NUL-ELEMENTU-----}
FOR I:=0 TO N DO FOR J:=0 TO W DO T[I,J]:=0;
{-----SAPOLNENIE MASSIWA-----}
FOR I:=1 TO N DO FOR J:=1 TO W DO
  IF J<P[I] THEN T[I,J]:=T[I-1,J]
  ELSE T[I,J]:=MAX(T[I-1,J],T[I-1,J-P[I]]+P[I]);
WRITELN('RESULTAT:',T[N,W] );
{-----KUCHA-----}
sum:=W;
for i:=n downto 1 do
  if t[i,sum]<>t[i-1,sum] then
    begin write (P[i]:3);
      sum:=sum-P[i]
    end;
end.

```