Поиск эвристики в задачи о разбиени кучи камней на две равного веса

Условие задачи

Дано множество камней разного веса. Разбросать их на две кучи максимально одинакового веса.

Задача имеет очевидное решение. У нас уже есть алгоритм построения всех сочетаний. Пусть первая куча — это очередное сочетание, а вторая куча — это то, что осталось в исходной куче. Тогда решение задачи сводится к построению сочетаний и простому запоминанию, какое из них дает наименьшую разность. Это решение исключительно неэффективно. Из N предметов можно построить 2^N сочетаний, что уже для N=10 дает 2048 сочетаний, для более существенного значения N количество сочетаний становится просто заоблачным. В общем, необходимо придумать более эффективный алгоритм, способный отсекать большое количество сочетаний. Можно пойти по пути построения «разумного метода», то есть такого, который вообще ничего не перебирает, а сразу строит то, что нужно. То есть речь идет о поиске закономерности. И здесь мы сталкиваемся с проблемой «отсутствия оснований для закономерности». Сущность проблемы применительно к нашей задаче звучит так. Закономерность ищется, исходя из какой-то известной информации, каких-то ключевых фактов, но про изначальную кучу камней ничего не известно, камни ведь собираются в нее произвольным образом, следовательно информации нет, а стало быть нет и закономерности.

Если вас эти рассуждения испугали, то это неверная реакция. Выявление проблемы помогает в поиске пути решения. Если проблема препятствует поискам, значит ее преодоление может дать путь и к решению задачи. Наше препятствие заключается в отсутствии информации о куче камней, следовательно, эту информацию надо добыть. Попробуем.

Единственно, что известно про камни, это то, что они имеют вес, и он может быть разным. Единственно, что может быть известно про два камня (если они отличимы по весу) — это то, что один из них вероятно тяжелее другого. А возможность сравнения двух камней по весу дает нам возможность разложить кучу на ряд камней по увеличению веса. Если это сделать, то перед нами будет уже не безликое нагромождение камней неопределенного веса, а упорядоченное множество, заключающее в себе хорошую закономерность. А теперь попробуем ее использовать.

В отсутствии желания перебирать все сочетаний, процедура раскладки камней будет выглядеть так: берем очередной камень и принимаем решение о том в какую кучу его положить, если удастся придумать, как только единожды принимать такое решение в отношении каждого камня, то вместо очень большого количества действий, мы выполним N простых. Поразмыслим о процедуре принятия решения.

Мы имеем упорядоченное множество камней, в программе это будет, видимо, массив чисел. Берем первый камень, так как множество упорядочено, это либо самый тяжелый, либо самый легкий, пусть для определенности это будет самый тяжелый. Положим его в левую кучу. Есть ли смысл следующий камень класть в эту же кучу. Может быть есть, но тогда мы весовой разрыв увеличиваем, более логично положить его в правую и так поступать до тех пор, пока правая куча не станет тяжелее левой, после чего по такому же правилу класть камни опять в левую и так до тех пор, пока камни не закончатся. Сказанное можно записать следующим алгоритмом:

```
Упорядочиваем множество камней по убыванию веса
Пока камни не закончились
  Берем очередной
```

```
Если вес куч различен, то
  Кладем его в кучу меньшего веса
  Иначе кладем его в левую
```

Алгоритм построен посредством вполне логичных рассуждений, если по нему написать программу, ее текст приведен далее, то можно для небольших тестовых примеров обнаружить, что она дает правильный результат, но означает ли это ее абсолютную правильность?

К сожалению, нет. Во-первых, примеры ничего не доказывают. Примером можно опровергнуть утверждение, но нельзя доказать, таков закон логики. Во-вторых, для пущей уверенности нам нужно просчитать достаточно большой пример, а в силу быстрого роста количества сочетаний, это практически невозможно. Что касается наших рассуждений, то при всей внешней логичности, они носят не строгий характер, и основываться на них нельзя. Выход из положения в четком строгом доказательстве достижимости нашим алгоритмом правильного результата. Первый шаг в поиске доказательства — просчет примеров. Это, во-первых, даст какую-то информацию о процессе и, во-вторых, чисто психологическую уверенность в том, что есть смысл искать доказательство. Возьмем следующую кучу камней: 18, 17, 13, 10, 10, 5, 4, 3, 3, 3, 1. Построим таблицу разложения.

Таблица. Разложение камней

Шаг	Левая куча	Сумма	Правая куча	Сумма
1	18	18		
2			17, 13	30
3	18,10,10	38		
4			17,13,5,4	39
5	18,10,10,3	41		
6			17,13,5,4,3	42
7	18,10,10,3,3	44		
8			17,13,5,4,3,1	43

Очевидно, что полученное разложение оптимально. Таких примеров можно построить много, мы ограничимся одним. Заметим, что разница между промежуточными суммами уменьшается. Это можно даже строго доказать, опершись на тот факт, что веса камней уменьшаются. Что это нам даст? Только то, что разности уменьшаются! А уменьшение величины абсолютно не гарантирует того, что величина достигнет возможного минимума.

Еще одна хорошая идея. Пусть дано множество чисел. Определим минимальную разницу, с которой это множество можно раскидать на два. А вторым шагом покажем, что при нашей стратегии именно минимальное значение и достигается. Новая идея споткнется о случайный характер исходного множества чисел. Математика умеет давать точные оценки, когда дело касается рядов чисел или последовательностей, но никто и никогда не оценивает количественно случайные последовательности, сама постановка этого вопроса не имеет смысла.

Иная идея. Давайте немного обдумаем постановку задачи. Предположим, мы завершили раскладку камней и получили некоторое решение. Далее нас интересует, нет ли более точного решения. Предположим, оно есть. Можно ли от нашей раскладки перейти к этой другой более точной? Конечно, можно. Для этого некоторую группу камней требуется переложить из левой кучи в правую и наоборот — другую группу надо переложить из правой кучи в левую. Итак, необходимо две группы камней поменять

местами. Оценить количество камней в них и их вес опять таки не получится, так как в самом общем случае они имеют самый общий состав, то есть случайный. Но допустим, что каждая из этих групп может состоять только из одного камня. То есть допустим, что если раскладка не оптимальна, то ее можно улучшить перемещением только двух камней — одного камня из левой кучи в правую и одного камня из правой кучи в левую. Если эта гипотеза верна, то для доказательства исходного утверждения будет достаточно доказать, что при выбранной стратегии раскладки камней в конечной ситуации нет пары камней, изменение положения которых ведет к уменьшению разности между весами левой и правой кучи.

Убедиться в разумности гипотезы о двух камнях можно опытным путем (хотя, конечно, это не доказательство). Возьмите достаточно большую кучу камней, половину камней положите вправо, половину влево, но так, чтобы разница в весах между ними была очевидно неоптимальной, и попробуйте уменьшать разницу перекладыванием либо одного, либо как максимум двух камней. Вы быстро убедитесь, что это действительно эффективно работает, но вот доказательство будет сложным, если оно вообще возможно.

Наше описание попыток доказательства оптимальности стратегии не заняло много места, но тем не менее, чтобы провести все эти выкладки, нужна серьезная интеллектуальная работа, а ее тупиковый характер должен навести на мысль об ошибочности теоремы. Если мы потратили много времени на ее доказательство, есть смысл потратить некоторое время на попытку ее опровержения. Опровержение иногда дело более простое. Наверное, потому что ложных утверждений неизмеримо больше, чем истинных. Кроме того, уже говорилось, что ничего нельзя доказать примером, а вот опровергнуть можно. Поищем контрпример. Не будем описывать процесс поиска, приведем его сразу. Исходная куча: 36, 25, 12, 10, 8, 7, 1. Наша стратегия дает такую раскладку:

Таблица. Разложение камней

Шаг	Левая куча	Сумма	Правая куча	Сумма
1	36	36		
2			25, 12	37
3	36, 10	46		
4			25,12,8,7	52
5	36,10,1	47		

Полученная разность равна 5. Перекладыванием двух камней эту разность можно уменьшить. А именно имеет место следующая раскладка (36, 12, 1) (25, 10, 8, 7) разность между ними равна 1.

Итак, красивая идея была неверна, но это не означает, что вся проделанная работа напрасна. Если бы вы искали опровергающий пример самостоятельно, то скорее всего заметили бы, что это непросто, и существует большой класс ситуаций, для которых наш красивый алгоритм работает хорошо. Если есть желание, то было бы полезно определить этот класс, если он очень велик, то красивый алгоритм имеет большой смысл, хотя и теряет универсальность. Во-вторых, наша работа очень важна для понимания технологии поиска доказательства. Общая схема всех наших попыток состояла из нескольких шагов, который мы опишем в виде правила.

Правило замены теоремы

- Заметим, что если некоторая гипотеза истинна, то исходная теорема сводится к другой более простой.
- Докажем эту гипотезу.
- Докажем теорему, к которой свелась исходная.

И хотя наша задача доказательства универсальности красивой идеи не решилась, все же приведем в листинг.

Листинг

```
program example;
uses crt;
var M: array[1..100] of integer; P: array[0..100] of 0..1;
    i, j, count, c1, c2, sum1, sum2, t: integer;
    f: boolean;
begin
 clrscr;
 readln(count);
 writeln;
 for i:=1 to count do
  readln(M[i]);
 writeln;
 for i:=count-1 downto 1 do
  for j:=1 to i do
   if M[j]<M[j+1] then
   begin
    t:=M[j]; M[j]:=M[j+1]; M[j+1]:=t;
   end;
 writeln;
 sum1:=0; sum2:=0; c1:=0; c2:=0;
 for i:=1 to count do
  if sum1>=sum2 then
  begin
   sum2:=sum2+M[i];
   c2:=c2+1;
  end
  else
  begin
   sum1:=sum1+M[i];
   c1:=c1+1;
  end;
 writeln(sum1,' ',sum2); writeln;
 readln;
end.
```

Утверждение о том, что упорядочение кучи камней дает необходимую для решения закономерность, оказалось надуманным. В какой-то степени это так. Но, во-первых, никто и не говорил, что вполне логичные рассуждения обязательно приводят к безусловной истине. К таковой истине не всегда приводят и математически строгие умозаключения, а мы рассуждали нестрого и, следовательно, могли ошибиться. Во-вторых, мы не совсем ошиблись, ведь решение все-таки получено, оно только не идеальное, а таковые решения

нужны не всегда. Иногда нужен метод, который может немного, но делает свою работу быстро. А именно такой метод мы и получили.

А теперь все же займемся идеальным решением задачи. Решение задачи полного перебора — это, конечно же, полный перебор. Никакой идеи, достойной серьезного обсуждения, у нас не будет, просто опишем набор технических действий.

Решение заключается в построении сочетаний. Сочетания мы уже получали с помощью представления двоичным числом. Алгоритм построим как цикл, внутри которого вычисляется очередное двоичное число. Если это число представляет, например, левую кучу, тогда правая куча — это то, что осталось. После построения левой и правой куч вычисляется разность их весов, и если эта разность меньше уже найденной, то запоминается новая разность и массивы левой и правой куч.

Листинг

```
program example;
 uses crt;
 var
  b,a,c1,c2:array[1..20] of word;
  n,i,s1,s2,min:word;
  q:boolean;
function proverka:boolean;
var
 q:boolean;
 i:word;
begin
 q:=false;
 for i:=1 to n do
  if b[i]=0 then q:=true;
 proverka:=q;
end;
procedure add_1;
var
 k,i:word;
begin
 i:=1;
 while (b[i]=1) and (i<=n) do i:=i+1;
 b[i]:=1;
 for k:=1 to i-1 do b[k]:=0;
end;
procedure mass;
var
 k:word;
begin
  for k:=1 to n do
   if b[k]=1 then
    begin
     c1[k]:=a[k];
     c2[k]:=0;
    end
   else
    begin
     c1[k]:=0;
     c2[k]:=a[k];
    end;
end;
begin
 clrscr;
 read(n);
 for i:=1 to n do
  begin
   read(a[i]);
   b[i]:=0;
```

```
  end;
 q:=true;
 while proverka do
  begin
   add_1;
   s1:=0;
   s2:=0;
   for i:=1 to n do
     if b[i]=0 then s1:=s1+a[i]
      else s2:=s2+a[i];
   if q then
    begin
     min:=abs(s1-s2);
     q:=false;
     mass;
    end
   else
     if abs(s1-s2)<min then
      begin
       min:=abs(s1-s2);
       mass;
      end;
  end;
  writeln('min=',min);
  for i:=1 to n do
   if c1[i]>0 then write(c1[i],' ');
  writeln;
  for i:=1 to n do
   if c2[i]>0 then write(c2[i],' ');
  writeln;
end.
```

В заключение

Попытка найти красивое решение частично провалилась. Иначе и быть не могло. Наши надежды на построение закономерности строились на песке. Закономерность вообще нельзя строить, ее можно только обнаружить. Это может оказаться весьма сложным и хитроумным делом. А рассмотренная задача все же переборная в чистом виде, но к нашей общей радости допускающей неточное, но быстрое решение. Такие решения позволяющие резко уменьшить потребность в переборе, еще называются эвристическими.