

## Задачи на генетическое программирование. Общие указания

Необходимо решить предлагаемые задачи, используя алгоритмы «генетического программирования». Вам необходимо самостоятельно определить, что является «хромосомой», «популяцией», как выполняется «скрещивание», «мутация» и «отбор лучших особей». В процессе реализации метода необходимо подобрать константы таким образом, чтобы добиться наилучшей скорости сходимости и точности метода.

Все задачи сформулированы в виде задач проверки свойств, то есть требующих ответ #f или #t в зависимости от того, существует ли *какое-либо* решение задачи, удовлетворяющее указанным свойствам. Однако в случае если решение существует, должно быть напечатано решение, оптимальное по некоторому параметру.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному, например, в виде графика минимума и максимума функции отбора в текущей популяции. Предусмотрите также возможность визуализации результата. Например, если задача сформулирована в терминах графов, визуализация результата может включать отображение графа на плоскости, в котором найденный подграф каким-либо образом выделен.

Номер своего варианта задачи Вы можете узнать на персональной страничке системы тестирования.

Решение задачи (исходный текст программы на scheme) должно находиться в единственном файле. Не разбивайте программу на модули!

# 1 Упаковка в контейнеры

На вход программе подаются два списка: список длины  $N$  ( $N > 0$ ) весов предметов  $W_i$  ( $W_i > 0, 1 \leq i \leq N$ ) и список длины  $K$  ( $K > 0$ ) вместимостей контейнеров  $C_j$  ( $C_j > 0, 1 \leq j \leq K$ ). Все веса и вместимости — целые числа.

Существует ли такое разбиение  $N$  предметов по  $K$  контейнерам, что для каждого предмета  $i$  есть контейнер  $j$ , в котором он размещён, и для каждого контейнера  $j$  вес всех предметов, в нём размещённых, не превышает  $C_j$ ? Если такое разбиение существует, напечатайте #t, затем количество использованных контейнеров, а затем само разбиение. Если разбиения не существует, напечатайте #f.

Из всех допустимых разбиений выберите разбиение, которое использует минимальное количество контейнеров. Разбиение печатайте в виде списка содержимого контейнеров. Содержимое контейнера — это список номеров размещённых в нём предметов.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

## Пример входных данных:

```
(1 2 2 3)
(4 4)
```

## Пример печати результата:

```
#t
2
((1 4) (2 3))
```

## 2 Задача о рюкзаке

На вход программе подаются два списка длины  $N$  ( $N > 0$ ) список весов предметов  $W_i$  ( $W_i > 0, i \leq 1 \leq N$ ) и список их стоимостей  $C_i$  ( $C_i > 0, i \leq 1 \leq N$ ). Все веса и стоимости предметов — целые числа. Затем на вход программе подаются два целых числа  $B$  ( $B > 0$ ) и  $K$  ( $K > 0$ ).

Существует ли такое подмножество предметов, для которого суммарный вес предметов в этом подмножестве  $W$  не превышает  $B$ , а суммарная стоимость предметов не меньше  $K$ ? Если такого подмножества не существует, напечатайте #f. Если такое подмножество существует, напечатайте #t, затем суммарный вес предметов, их суммарную стоимость, а затем список номеров предметов, вошедших в это подмножество. Из всех возможных разбиений выберите такое, которое максимизирует суммарную стоимость предметов в рюкзаке.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

### Пример входных данных:

```
(1 1 2 2)
(1 2 3 4)
3
4
```

### Пример печати результата:

```
#t
3
4
(2 4)
```

### 3 Хроматическое число графа

На вход программы подаётся список рёбер неориентированного графа  $G = \langle V, E \rangle$ . Вершины обозначаются атомами, рёбра представляются списками вида

```
( NODE1 NODE2 )
```

Затем на вход программы подаётся целое число  $K$  ( $1 \leq K$ ). Можно ли раскрасить граф  $G$  не более чем в  $K$  цветов, то есть существует ли функция  $c : V \rightarrow \{1 \dots K\}$  такая, что если  $(u, v) \in E$ , то  $c(u) \neq c(v)$ ? Если раскраски графа не существует, напечатайте #f. Если раскраска графа существует, то сначала напечатайте #t, затем количество цветов, в которые раскрашен граф, а затем список цветов вершин. Элементами этого списка являются списки из двух элементов следующего вида

```
( NODE COLOR )
```

Из всех возможных вариантов раскраски выберите тот, который требует минимального числа цветов.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

**Пример входных данных:**

```
((a b) (b c) (c d) (a d))  
2
```

**Пример печати результата:**

```
#t  
2  
((a 1) (b 2) (c 1) (d 2))
```

## 4 Доминирующее множество

На вход программы подаётся список рёбер неориентированного графа  $G = \langle V, E \rangle$ . Вершины обозначаются атомами, рёбра представляются списками вида

```
( NODE1 NODE2 )
```

Затем на вход программы подаётся целое число  $K$  ( $1 \leq K$ ).

Существует ли в графе  $G$  доминирующее подмножество вершин, то есть такое подмножество вершин  $V' \subseteq V$ , что  $|V'| \leq K$  и каждая вершина  $v$  не из этого подмножества вершин ( $v \in V - V'$ ) соединена ребром по крайней мере с одной вершиной из множества  $V'$ ? Если доминирующего подмножества вершин не существует, напечатайте #f. Если доминирующее подмножество существует, напечатайте #t, затем количество вершин в доминирующем подмножестве, а затем список вершин, входящих в доминирующее подмножество. Из всех возможных вариантов доминирующего подмножества выберите вариант с минимальным количеством вершин.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

**Пример входных данных:**

```
((a b) (a c) (a d))  
1
```

**Пример печати результата:**

```
#t  
1  
(a)
```

## 5 Задача коммивояжёра

На вход программе подаётся описание взвешенного неориентированного графа  $G = \langle V, E \rangle$  в виде списка описаний рёбер. Каждое описание ребра имеет вид

```
( NODE1 NODE2 WEIGHT )
```

где NODE1, NODE2 — атомы, WEIGHT — целое положительное число. Затем на вход программе подаётся целое положительное число  $K$ .

Существует ли в графе  $G$  гамильтонов цикл, то есть путь, начинающийся и заканчивающийся в одной и той же вершине и проходящий через каждую вершину только один раз, стоимости не более чем  $K$ ? Стоимость пути — это суммарная стоимость рёбер, составляющих путь. Если гамильтонов цикл не существует, напечатайте #f. Если гамильтонов цикл существует, напечатайте #f, далее стоимость цикла, а затем гамильтонов путь в виде списке. Первым и последним элементом списка должен быть один и тот же атом. Если существует несколько возможных гамильтоновых путей, найдите путь с минимальной стоимостью.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

### Пример входных данных:

```
((a b 1) (b c 1) (c d 1) (a d 1))  
4
```

### Пример печати результата:

```
#t  
4  
(a b c d a)
```

## 6 Клика

На вход программе подаётся описание неориентированного невзвешенного графа  $G = \langle V, E \rangle$  в виде списка описаний рёбер. Вершины обозначаются атомами, рёбра представляются списками вида

```
( NODE1 NODE2 )
```

Затем на вход программе подаётся целое число  $K$  ( $1 \leq K \leq |V|$ ).

Существует ли в графе  $G$  клика мощности, не меньшей  $K$ ? Кликой графа называется полный подграф графа. Мощность клики — это количество вершин в ней. Если такой клики не существует, напечатайте #f. Если клика существует, напечатайте #t, затем мощность клики, затем список вершин, входящих в клику. Из всех возможных решений задачи, удовлетворяющих условию, выберите решение с максимальной мощностью клики.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

### Пример входных данных:

```
((a b) (b c) (c d) (a d) (a c))  
3
```

### Пример печати результата:

```
#t  
3  
(a b c)
```

## 7 Прямоугольное сжатие картинки

На вход программы подаётся квадратная 0–1 матрица  $M$  размера  $N \times N$ . Матрица вводится в виде списка из  $N$  строк матрицы, где каждая строка — это список из  $N$  элементов матрицы. Затем задаётся положительное целое число  $K$ .

Можно ли покрыть все единичные элементы матрицы  $M$  не более чем  $K$  прямоугольниками? Другими словами, существует ли такая последовательность четвёрок  $(a_i, b_i, c_i, d_i)$  ( $1 \leq i \leq K$ ,  $1 \leq a_i, b_i, c_i, d_i \leq N$ ,  $a_i \leq c_i$ ,  $b_i \leq d_i$ ), в которой  $a_i, b_i$  — это номера строк матрицы, а  $b_i, d_i$  — номера столбцов. Эта последовательность должна удовлетворять условию, что любой единичный элемент матрицы должен покрываться хотя бы одним прямоугольником, и любой прямоугольник должен состоять только из единичных элементов матрицы. Если такого покрытия не существует, напечатайте #f. Если покрытие существует, напечатайте #t, затем количество прямоугольников, а затем список описаний прямоугольников. Каждое описание прямоугольника представляет собой список из 4 элементов как описано выше.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

### Пример входных данных:

```
((1 1 0 0) (1 1 0 0) (0 0 1 1) (0 0 1 1))
2
```

### Пример печати результата:

```
#t
2
((1 1 2 2) (3 3 4 4))
```



## 8 Разрезание циклов вершинами

На вход программе подаётся описание невзвешенного ориентированного графа  $G = \langle V, E \rangle$  в виде списка рёбер. Вершины графа представлены атомами, а описание каждого ребра имеет вид

```
( FROM TO )
```

Затем на вход программе подаётся число  $K$

Существует ли в графе  $G$  такое подмножество вершин  $V'$  ( $V' \subseteq V$ ) мощности не большей  $K$ , что каждый цикл в графе  $G$  содержит хотя бы одну вершину из множества  $V'$ ? Если такого подмножества вершин не существует, напечатайте  $\#f$ , а если такое подмножество вершин существует, напечатайте  $\#t$ , затем количество вершин в нём, а затем список вершин, входящих в это подмножество. Из всех допустимых подмножеств выберите подмножество минимальной мощности.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

**Пример входных данных:**

```
(( a b) (b c) (c d) (d a) )
```

```
1
```

**Пример печати результата:**

```
#t
```

```
1
```

```
(a)
```

## 9 Разрезание циклов рёбрами

На вход программе подаётся описание невзвешенного ориентированного графа  $G = \langle V, E \rangle$  в виде списка рёбер. Вершины графа представлены атомами, а описание каждого ребра имеет вид

```
( FROM TO )
```

Затем на вход программе подаётся число  $K$

Существует ли в графе  $G$  такое подмножество рёбер  $E' \subseteq E$  мощности не большей  $K$ , что каждый цикл в графе  $G$  содержит хотя бы одно ребро из множества  $V'$ ? Если такого подмножества рёбер не существует, напечатайте #f, а если такое подмножество рёбер существует, напечатайте #t, затем количество вершин в нём, а затем список рёбер, входящих в это подмножество. Из всех допустимых подмножеств выберите подмножество минимальной мощности.

В процессе нахождения решения должна быть предусмотрена визуализация процесса приближения текущего решения задачи к оптимальному. Предусмотрите возможность визуализации результата.

**Пример входных данных:**

```
(( a b) (b c) (c d) (d a))
```

```
1
```

**Пример печати результата:**

```
#t
```

```
1
```

```
(( a b))
```