# **Жадные алгоритмы**

## **Задача о составлении расписания**

Допустим, имеется учебный класс, в котором нужно провести как можно больше уроков. У вас есть список уроков.

Таблица 1

|  |  |  |
| --- | --- | --- |
| **Урок** | **Время начала** | **Время окончания** |
| Рисование | 09:00 | 10:00 |
| Английский язык | 09:30 | 10:30 |
| Математика | 10:00 | 11:00 |
| Информатика | 10:30 | 11:30 |
| Музыка | 11:00 | 12:00 |

Провести в классе все уроки не получится, потому что некоторые из них пересекаются по времени.

Вопросы.

1. Как отобрать уроки, чтобы полученный набор оказался самым большим из возможных?
2. Можете ли вы предложить свой вариант расписания?
3. Можете ли вы предложить свой алгоритм для решения такого рода задач?

Напомним, что решение задачи (то есть алгоритм) должен работать при всех возможных вариантах списка уроков.

На самом деле алгоритм оказывается крайне простым.

1. Выбрать урок, завершающийся раньше всех. Это первый урок, который будет проведен в классе.
2. Затем выбирается урок, начинающийся после завершения первого урока. И снова следует выбрать урок, который завершается раньше всех.

Получите ответ для приведенного выше списка уроков и составьте расписание для учебного кабинета.

Ответ: рисование, математика, музыка.

Вопросы:

1. Кажется ли вам, что этот алгоритм слишком прост?
2. Если он так очевиден, может быть, он неправильный?

Только что рассмотренный пример – это пример **жадного** алгоритма. Жадные алгоритмы очень просты. На каждом шаге жадный алгоритм просто выбирает оптимальный вариант. В технической терминологии: на каждом шаге выбирается **локально**-оптимальное решение, а в итоге вы получаете **глобально**-оптимальное решение.

Конечно, жадные алгоритмы работают не всегда. Но их реализация крайне проста. Чтобы убедиться в этом, попробуйте решить задания 3.2. – 3.5 из рабочего листа.

Составьте свой список дел (уроков, мероприятий, маршрутов) с указанием времени начала и времени окончания мероприятий. Составьте расписание, воспользовавшись жадным алгоритмом.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## **Задача о рюкзаке**

«Задача о рюкзаке» описывает ситуацию, когда кто-то (главный герой задачи) собирается в длительный поход. Обычно с собой хочется взять много разных любимых вещей, без которых сложно представить свою жизнь. Кроме того, нужно взять необходимые именно в походе вещи и одежду. Однако ёмкость рюкзака всегда ограничена.

Другими словами: перед вами набор вещей разной степени важности (стоимости, ценности), которые вы хотите взять с собой в поход помимо жизненно необходимых вещей. И объём вашего рюкзака ограничен. Требуется подобрать такой набор вещей **максимальной** важности (стоимости, ценности), которые можно уложить в рюкзак.

Вопросы.

1. Как определить важность вещей?
2. Как ранжировать этот список?
3. Предложите свой избыточный набор вещей. Соберите свой походный рюкзак.
4. Предложите свой вариант алгоритма решения данной задачи. Напомним, что решение задачи (то есть алгоритм) должен работать при всех возможных вариантах.

Алгоритм выглядит следующим образом:

1. Выбрать самый дорогой предмет, который поместится в рюкзаке.
2. Выбрать следующий по стоимости предмет, который поместится в рюкзаке.
3. И так далее...

Но рассмотрим такой пример.

Таблица 2

|  |  |  |
| --- | --- | --- |
| **Предмет** | **Вес, кг** | **Ценность, усл. ед.** |
| Музыкальная колонка | 0.7 | 100 |
| Ноутбук | 1.8 | 250 |
| Набор «Юный натуралист» | 1 | 175 |

В рюкзаке осталось немного места для предметов общим весом не более 2 кг. Самый дорогой предмет – ноутбук. Если взять его, для других предметов места не останется.

Но можно взять музыкальную колонку и набор «юный натуралист» (общий вес не превышает 2 кг). Общая ценность двух этих предметов превышает ценность ноутбука.

Очевидно, жадная стратегия не дает **оптимального** решения. Впрочем, результат не так уж далек от оптимального. Кроме того, вам может подойти и первый вариант решения.

В некоторых случаях для решения задачи достаточно алгоритма, способного решить задачу пусть не идеально, но достаточно хорошо. Особенно если его реализация крайне проста. Как раз в таких случаях жадные алгоритмы подходят просто идеально.

Существует и другие условия задачи о рюкзаке. И другие решения задачи о рюкзаке. Одним из них является решение с помощью динамического программирования. Его рассмотрение выходит за рамки нашего урока, но очень советуем разобраться с ним самостоятельно.

Составьте свой список вещей с указанием веса и ценности каждого из них, которые вы бы взяли с собой в длительный поход или путешествие. Самостоятельно ограничьте максимальный вес своего «рюкзака». Попробуйте решить задачу, воспользовавшись жадным алгоритмом.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

##