Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

Выполнил студент группы КС-36 Алёшин Михаил Алексеевич

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/MAAleshin\_36\_algo

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: 14.02.2022

Оглавление

[Описание задачи. 2](#_Toc63548272)

[Описание метода/модели. 2](#_Toc63548273)

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

У вас есть несколько камней известного веса W1, …, Wn. Напишите программу, которая распределит камни в две кучи так, что разность весов этих двух куч будет минимальной. Необходимо создать программу которая будет генерировать от 1 до 20 камней с различными весами от 1 до 10000. Итоговый набор камней должен подаваться в функцию раскладки и в ответ она должна выдавать разницу разделения общей кучи на две.

# Описание метода/модели.

Динамическое программирование чем-то похоже на жадные алгоритмы, в частности тем, что как и в случае с жадным алгоритмом здесь оперируют подзадачами и их решением. Суммарно идея динамического программирования заключается в том, что мы разбиваем задачу на малые подзадачи, спускаясь вниз, далее, находим самую минимальную подзадачу, решаем, ее, затем, имея ее решение, мы создаем подзадачи, которые включают в свое решение решенную более меньшую подзадачу, снова получаем их решения, сохраняем их, и движемся далее вверх.

По итогу динамическое программирование требует следующего для решения какой-либо задачи:

* перекрывающиеся подзадачи;
* оптимальная подструктура;
* возможность запоминания решения часто встречающихся подзадач.

Словами общую идею построения алгоритма можно описать следующим образом:

* Разбиение задачи на подзадачи меньшего размера.
* Нахождение оптимального решения подзадач рекурсивно, проделывая такой же трехшаговый алгоритм.
* Использование полученного решения подзадач для конструирования решения исходной задачи.

Есть 2 варианта динамического программирования:

* нисходящее динамическое программирование: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. Используется запоминание для решений уже решенных подзадач.
* восходящее динамическое программирование: все подзадачи, которые впоследствии понадобятся для решения исходной задачи просчитываются заранее и затем используются для построения решения исходной задачи. Этот способ лучше нисходящего программирования в смысле размера необходимого стека и количества вызова функций, но иногда бывает нелегко заранее выяснить, решение каких подзадач нам потребуется в дальнейшем.

# Выполнение задачи.

Реализована функция для работы DP(), в которой производятся расчеты для первой кучи камней. Сначала список dp заполняется значениями: сможем ли мы заполнить рюкзак полностью. Если нет, какой максимальный вес мы можем набрать? Например, для списка [1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0], для рюкзака вместимостью 14, максимальное значение будет 13. Далее по списку dp производится восстановление ответа, то есть резульат, какие камни могут поместиться в данную кучу (сумку). После этого оставшиеся камни добавляются в кучу и проверяется разность весов этих двух куч. Вторая куча будет так же оптимальна, так как её вес будет тоже стремиться к (вес общей кучи)/2. Далее посмотрим на несколько сгенерированных случаев:

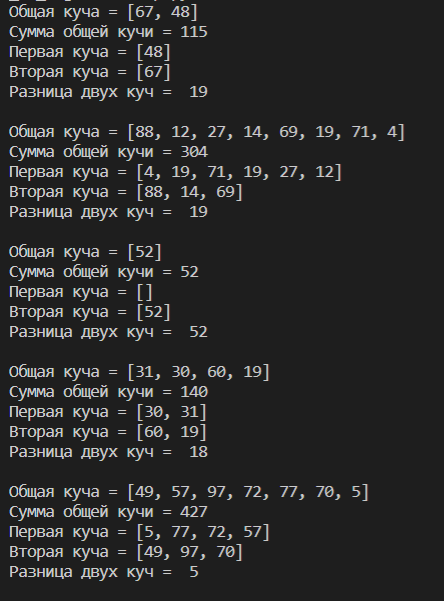


Рис. 1. Результат работы алгоритма.

Далее попробуем забить решение на проверку:

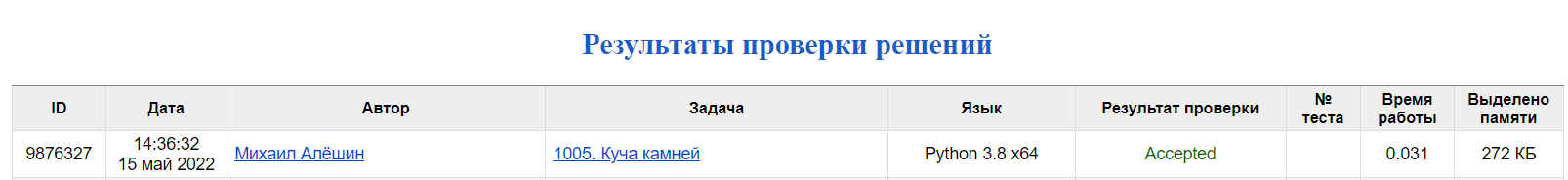


Рис. 2. Результат проверки решения.

# Заключение.

Динамическое программирование зачастую быстрее решает задачу путем запоминания вычислений на предыдущей итерации, что иногда позволяет привести рекурсивную задачу к итеративной. Однако, в случае динамического программирования приходится тратить много памяти для хранения промежуточной информации. Это и есть основная суть динамического программирования – трата памяти взамен на время. Однако динамическое программирование в некоторых случаях может дать неверный ответ (крайне редко), и поэтому для самого точного решения в данной задаче будет необходимо применить метод перебора.