Лабораторная работа 7 - Динамическое программирование

Цель

Разобрать базовые приёмы динамического программирования и применить их в задачах.

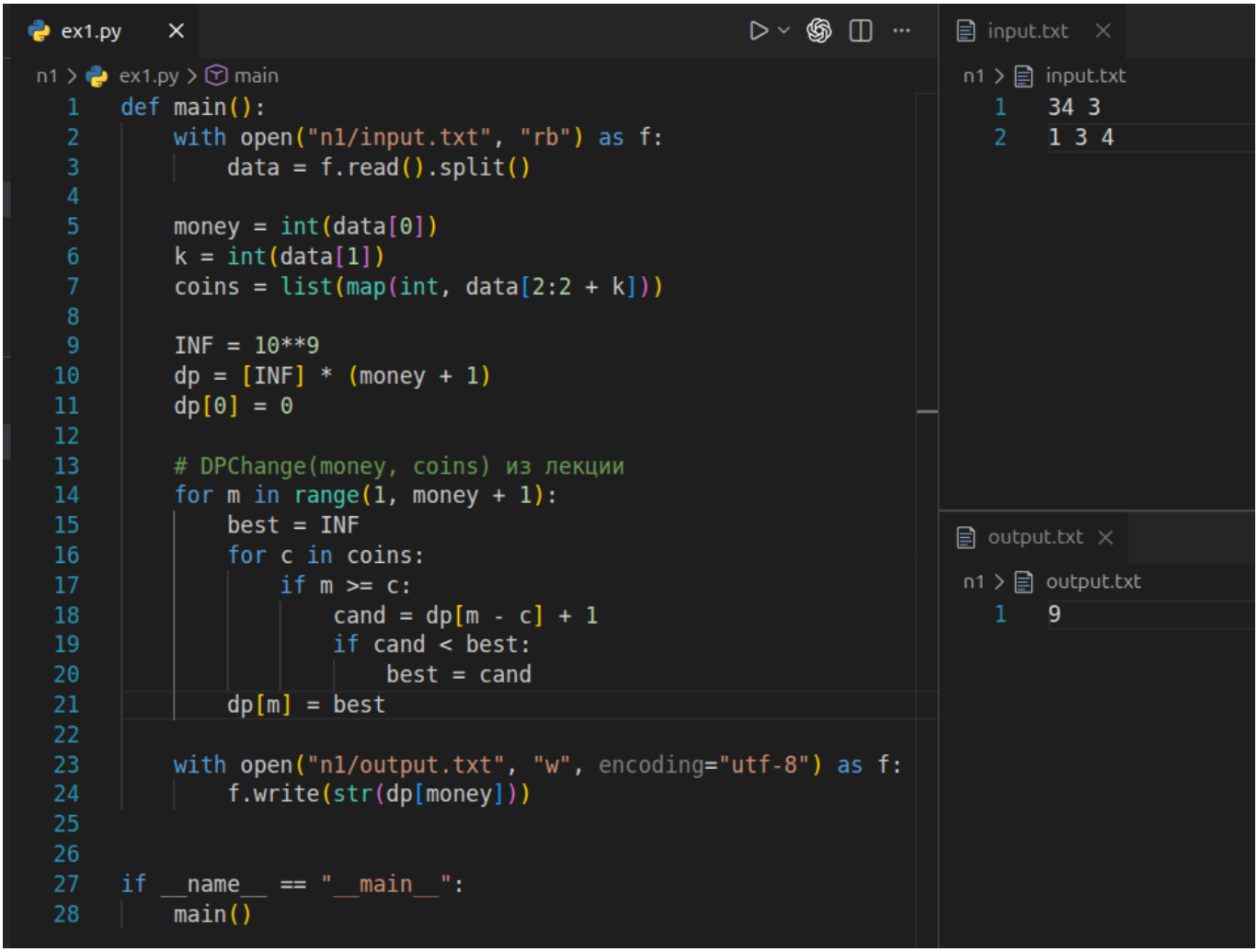
Задания и решения

Задание 1 - Обмен монет (DPChange)

Нужно найти минимальное количество монет, чтобы набрать сумму `money`.

Сделано:

- `dp[m]` - минимум монет для суммы `m`
- переход: `dp[m] = min(dp[m - coin] + 1)` по всем `coin`
- заполняем `dp` снизу вверх от `0` до `money`



Проверено на нескольких input-файлах (разные наборы монет, маленькие и большие суммы) - работает корректно.

Задание 2 - Примитивный калькулятор

Нужно из 1 получить `n` с минимальным числом операций (`+1`, `*2`, `*3`) и вывести:

- минимальное число операций
- саму последовательность чисел

Сделано:

- `dp[i]` - минимум операций до `i`
- `prev[i]` - откуда пришли в `i`, чтобы восстановить путь
- для каждого `i` сравниваем варианты: `i-1`, `i/2` (если делится), `i/3` (если делится)
- по `prev[]` восстанавливаем ответ назад от `n` до `1`

```
from array import array


def main():
    with open("n2/input.txt", "rb") as f:
        data = f.read().split()

    n = int(data[0])

    dp = array("I", [0]) * (n + 1) # dp[i] - минимум операций до i
    prev = array("I", [0]) * (n + 1) # prev[i] - откуда пришли в i

    for i in range(2, n + 1):
        best = dp[i - 1] + 1
        p = i - 1

        if i % 2 == 0:
            cand = dp[i // 2] + 1
            if cand < best:
                best = cand
                p = i // 2

        if i % 3 == 0:
            cand = dp[i // 3] + 1
            if cand < best:
                best = cand
                p = i // 3

        dp[i] = best
        prev[i] = p

    seq = []
    cur = n
    while True:
        seq.append(cur)
        if cur == 1:
            break
        cur = prev[cur]
```

```
    seq.reverse()

    out = [str(len(seq) - 1), " ".join(map(str, seq))]
    with open("n2/output.txt", "w", encoding="utf-8") as f:
        f.write("\n".join(out))


if __name__ == "__main__":
    main()
```

Проверено на нескольких input-файлах (маленькие n, большие n, случаи с несколькими оптимальными путями) - вывод корректный.