

Федеральное государственное автономное образовательное учреждение
высшего образования
**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО**
Факультет систем управления и робототехники

Лабораторная работа №7
Задачи №1162, 1806

Студент: Сайфуллин Д.Р.
Поток: АиСД R23 1.3
Преподаватель: Тропченко А.А.

Санкт-Петербург
2025 г.

Задача №1162. Currency Exchange

В городе работает несколько обменных пунктов, каждый из которых специализируется ровно на двух валютах и принимает/выдаёт только их. Каждый пункт задаётся шестью числами: A , B , R_{AB} , C_{AB} , R_{BA} , C_{BA} , где

- A и B — номера двух валют, которыми оперирует пункт,
- R_{AB} — курс обмена из валюты A в валюту B (сколько единиц B вы получите за 1 единицу A),
- C_{AB} — комиссия в валюте A при обмене $A \rightarrow B$,
- R_{BA} и C_{BA} — аналогично для обмена $B \rightarrow A$.

Ник располагает суммой V в валюте S и хочет узнать, может ли он, совершая любую последовательность обменных операций (не повторяя один и тот же пункт более одного раза), увеличить свой капитал в валюте S (то есть получить сумму, строго большую V). Во всех промежуточных операциях сумма должна оставаться неотрицательной.

Исходные данные:

Первая строка содержит четыре числа: N M S V , где N ($1 \leq S \leq N \leq 100$) — число различных валют, M ($1 \leq M \leq 100$) — число обменных пунктов, S — номер начальной валюты, V — начальный объём в валюте S ($0 \leq V \leq 10^3$). Далее идёт M строк, каждая из которых содержит шесть чисел A B R_{AB} C_{AB} R_{BA} C_{BA} , где $0 \leq C_{AB}, C_{BA} \leq 100$, $10^{-2} \leq R_{AB}, R_{BA} \leq 10^2$ с точностью не хуже двух знаков после десятичной точки.

Результат:

Если Ник может увеличить свой капитал в валюте S (то есть получить больше V), выведите YES, в противном случае выведите NO.

Рабочий код

```
1 def solve():
2     n, m, s, v = input().split()
3     n, m, s = int(n), int(m), int(s)
4     v = float(v)
5
6     edges = []
7     for _ in range(m):
8         a, b, rab, cab, rba, cba = input().split()
9         a, b = int(a), int(b)
10        rab, cab = float(rab), float(cab)
11        rba, cba = float(rba), float(cba)
12        edges.append((a, b, rab, cab))
13        edges.append((b, a, rba, cba))
14
15    value = [0.0] * (n + 1)
16    value[s] = v
17
18    for _ in range(n - 1):
19        updated = False
20        for u, w, rate, comm in edges:
21            if value[u] >= comm:
22                new_amount = (value[u] - comm) * rate
```

```

23         if new_amount > value[w]:
24             value[w] = new_amount
25             updated = True
26     if not updated:
27         break
28
29     for u, w, rate, comm in edges:
30         if value[u] >= comm and (value[u] - comm) * rate > value[w]:
31             print("YES")
32             return
33
34     print("NO")
35
36
37 if __name__ == "__main__":
38     solve()

```

Объяснение алгоритма

Представляем валюты вершинами, а каждый обмен $u \rightarrow v$ с курсом r и комиссией c — ребром, дающим $\text{value}_v = (\text{value}_u - c)r$. Запускаем Bellman–Ford на $N - 1$ итерацию релаксации; затем дополнительная итерация: если какое-то ребро улучшает значение, значит есть выгодный цикл — выводим «YES», иначе «NO».

Задача №1806. Мобильные телеграфы

Каждому бойцу 25-й стрелковой дивизии выдали мобильный телеграф с десятизначным номером. Передача сообщения с телеграфа A на телеграф B возможна лишь в том случае, если из номера A можно получить номер B либо

- изменив ровно одну цифру в некоторой позиции, или
- поменяв местами две цифры между собой.

Время передачи зависит от длины наибольшего общего префикса их номеров: если первые k цифр совпадают, а $(k + 1)$ -я различается, то передача идёт за t_k секунд (где $0 \leq k \leq 9$).

Анка увидела группу белых за спинами красноармейцев и хочет передать Чапаеву сообщение. Номер телеграфа Анки указан первым в списке, Чапаева - последним. Требуется найти маршрут пересылки, который минимизирует общее время передачи от первого телеграфа к последнему.

Исходные данные:

В первой строке дано целое n ($2 \leq n \leq 50000$) - число бойцов (телеграфов) в дивизии. Во второй строке через пробел записаны десять целых чисел t_0, t_1, \dots, t_9 ($1 \leq t_i \leq 10000$) - время передачи при совпадении первых i цифр. В каждой из следующих n строк дано по одной десятизначной строке — номера телеграфов, выданных бойцам. Все номера попарно различны. Первая строка соответствует Анке, последняя - Чапаеву.

Результат:

Если доставить сообщение нельзя, выведите единственное число -1. Иначе в первой строке выведите минимальное время доставки. Во второй строке - число k бойцов (вершин), через которых проходит сообщение, включая Анку и Чапаева. В третьей строке - k целых чисел (от 1 до n), номера бойцов в порядке маршрута.

Рабочий код

```
1 import heapq
2
3 def main():
4     n = int(input())
5     times = list(map(int, input().split()))
6     ids = [input().strip() for _ in range(n)]
7     index = {ids[i]: i for i in range(n)}
8
9     INF = 10**18
10    dist = [INF]*n
11    prev = [-1]*n
12    dist[0] = 0
13
14    pq = [(0, 0)]
15    while pq:
16        d, u = heapq.heappop(pq)
17        if d > dist[u]:
18            continue
19        if u == n-1:
20            break
21
22        s = ids[u]
23        for i in range(10):
```

```

24         cost = times[i]
25         orig = s[i]
26         for c in '0123456789':
27             if c == orig:
28                 continue
29             t = s[:i] + c + s[i+1:]
30             v = index.get(t)
31             if v is not None:
32                 nd = d + cost
33                 if nd < dist[v]:
34                     dist[v] = nd
35                     prev[v] = u
36                     heapq.heappush(pq, (nd, v))
37         for j in range(i+1, 10):
38             if s[j] == orig:
39                 continue
40             lst = list(s)
41             lst[i], lst[j] = lst[j], lst[i]
42             t = ''.join(lst)
43             v = index.get(t)
44             if v is not None:
45                 nd = d + cost
46                 if nd < dist[v]:
47                     dist[v] = nd
48                     prev[v] = u
49                     heapq.heappush(pq, (nd, v))
50
51     if dist[n-1] == INF:
52         print(-1)
53     else:
54         path = []
55         cur = n-1
56         while cur != -1:
57             path.append(cur+1)
58             cur = prev[cur]
59         path.reverse()
60         print(dist[n-1])
61         print(len(path))
62         print(*path)
63
64 if __name__ == "__main__":
65     main()

```

Объяснение алгоритма

Каждый номер — вершина графа, переходы между ними возможны двумя способами:

- изменить одну цифру (положение первого отличия i) или
- поменять местами две цифры (первая позиция различия также i).

Вес такого ребра равен t_i . Далее запускаем алгоритм Дейкстры от вершины 1 до n -й, восстанавливаем путь через массив предков. Если n -я вершина недостижима, выводим -1 .

Статус проверки

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10957761	16:04:10 13 май 2025	Dinislam	1806. Мобильные телеграфы	PyPy 3.10 x64	Accepted		2.093	22 300 КБ
10957756	15:47:49 13 май 2025	Dinislam	1162. Currency Exchange	PyPy 3.10 x64	Accepted		0.109	1 480 КБ

Рис. 1: Результат проверки