





import tkinter as tk

from tkinter import ttk, messagebox

import numpy as np

def northwest\_corner(costs, supply, demand):

    supply = np.array(supply)

    demand = np.array(demand)

    if sum(supply) != sum(demand):

        return None, f"Задача не сбалансирована. Сумма поставок ({sum(supply)}) не равна сумме потребностей ({sum(demand)})."

    num\_suppliers = len(supply)

    num\_consumers = len(demand)

    allocation = np.zeros((num\_suppliers, num\_consumers), dtype=int)

    total\_cost = 0

    i, j = 0, 0

    while i < num\_suppliers and j < num\_consumers:

        quantity = min(supply[i], demand[j])

        allocation[i, j] = quantity

        total\_cost += costs[i, j] \* quantity

        supply[i] -= quantity

        demand[j] -= quantity

        if supply[i] == 0:

            i += 1

        else:

            j += 1

    return allocation, total\_cost

def solve\_and\_display():

    try:

        costs\_str = costs\_text.get("1.0", "end-1c")

        supply\_str = supply\_entry.get()

        demand\_str = demand\_entry.get()

        costs = np.array([list(map(int, row.split())) for row in costs\_str.splitlines()])

        supply = list(map(int, supply\_str.split()))

        demand = list(map(int, demand\_str.split()))

        result = northwest\_corner(costs, supply, demand)

        if result[0] is not None:

            allocation, total\_cost = result

            result\_text.config(state=tk.NORMAL)

            result\_text.delete(1.0, tk.END)

            result\_text.insert(tk.END, "Базисный план (метод северо-западного угла):\n")

            for row in allocation:

                result\_text.insert(tk.END, str(row) + "\n")

            result\_text.insert(tk.END, f"\nОбщие затраты: {total\_cost}")

            result\_text.config(state=tk.DISABLED)

        else:

             messagebox.showerror("Ошибка", result[1])

             result\_text.config(state=tk.NORMAL)

             result\_text.delete(1.0, tk.END)

             result\_text.config(state=tk.DISABLED)

    except ValueError:

        messagebox.showerror("Ошибка", "Некорректный формат ввода")

        result\_text.config(state=tk.NORMAL)

        result\_text.delete(1.0, tk.END)

        result\_text.config(state=tk.DISABLED)

root = tk.Tk()

root.title("Транспортная задача")

ttk.Label(root, text="Матрица затрат: (строки через Enter, числа через пробел)").grid(row=0, column=0, sticky="w", padx=5, pady=5)

costs\_text = tk.Text(root, height=5, width=40)

costs\_text.grid(row=0, column=1, padx=5, pady=5)

ttk.Label(root, text="Поставки (через пробел):").grid(row=1, column=0, sticky="w", padx=5, pady=5)

supply\_entry = ttk.Entry(root, width=40)

supply\_entry.grid(row=1, column=1, padx=5, pady=5)

ttk.Label(root, text="Потребности (через пробел):").grid(row=2, column=0, sticky="w", padx=5, pady=5)

demand\_entry = ttk.Entry(root, width=40)

demand\_entry.grid(row=2, column=1, padx=5, pady=5)

solve\_button = ttk.Button(root, text="Решить", command=solve\_and\_display)

solve\_button.grid(row=3, column=0, columnspan=2, pady=10)

ttk.Label(root, text="Результат:").grid(row=4, column=0, sticky="w", padx=5, pady=5)

result\_text = tk.Text(root, height=10, width=60, state=tk.DISABLED)

result\_text.grid(row=4, column=1, padx=5, pady=5)

root.mainloop()