DS 23 – Problem, Probleminstanz und abstrahierte Darstellung

Informatik 10 – MINT-Zweig | Informatisches Problemlösen

Lernziel

Ich unterscheide zwischen einem Problem und einer Probleminstanz und untersuche kleine, abstrahiert dargestellte Probleminstanzen, um strukturelle Eigenschaften zu erkennen.

> **Hinweis:** In diesem Themenfeld steht laut Lehrplan die **Auseinandersetzung mit Problemstrukturen und Lösungsstrategien** im Vordergrund – die Umsetzung in einer Programmiersprache ist **nicht** vorgesehen. Python-Zellen in diesem und den folgenden Notebooks sind daher **optional** und dienen nur zur Veranschaulichung.

🕘 Stundenverlaufsplan (Doppelstunde, 90 Minuten)

Orientierungsrahmen für 11 SuS: Die ersten 80 Minuten bilden den Kern. Die letzten 10 Minuten sind bewusst flexibel und werden nicht mit neuem Pflichtstoff belegt.

Phase	Zeit	Inhalt/Aktion	Sozialform
Einstieg und Aktivierung	10'	Ein allgemeines Routenproblem und eine konkrete Karte gegenüberstellen.	Plenum; kurze Einzeldenkzeit
Fragend-entwickelnde Erarbeitung	25'	Problem, Probleminstanz und abstrahierte Darstellung aus den Unterschieden entwickeln.	Unterrichtsgespräch; Beiträge der SuS werden sichtbar festgehalten
Gemeinsame Umsetzung und Sicherung	20'	Die TSP-Instanz gemeinsam als Tabelle und gedanklich als Graph untersuchen.	Plenum; Live-Coding, Tafelbild oder gemeinsame Simulation
Kurzer Transfer	15'	Beispiele als Problem oder Probleminstanz klassifizieren und begründen.	Einzelarbeit mit anschließendem Tandemabgleich (4 Tandems + 1 Trio)
Auswertung	10'	Symmetrie der Entfernungstabelle und ihre Bedeutung erläutern lassen.	Plenum; Lösung erklären und Fachsprache sichern
Reserve / Vertiefung	10'	Eigene Instanz formulieren; Python-Liste nur als optionale Demonstration.	Flexibel; bei Bedarf für Fragen, Fehlersuche oder Abschluss nutzen

🗣️ Sprachhilfe

- „Es handelt sich um ein Problem/eine Probleminstanz, weil …“
- „Der Ansatz prüft/wählt … und benötigt dabei …“
- „Die Heuristik ist hier geeignet/nicht geeignet, weil …“

📌 Merksätze

- Ein **Problem** ist eine allgemeine Fragestellung mit "Leerstellen" (z. B. "Finde die kürzeste Route zwischen beliebigen Städten").
- Eine **Probleminstanz** ist eine konkrete Ausprägung dieses Problems mit festen Werten (z. B. "Finde die kürzeste Route zwischen genau diesen 5 gegebenen Städten mit diesen Entfernungen").

- Probleminstanzen werden häufig **abstrahiert dargestellt**, um sie handhabbar zu machen, z. B. als:
  - **Tabelle** (z. B. Entfernungstabelle zwischen Orten)
  - **Graph** (Knoten = Objekte, Kanten = Beziehungen/Entfernungen)
  - **Liste** (z. B. Liste von Gegenständen mit Gewicht und Wert)
- Das Untersuchen mehrerer **Varianten** einer Probleminstanz hilft, **Symmetrien** und Muster zu erkennen, die spätere Lösungsstrategien vereinfachen können.

💡 Beispiel: Problem des Handlungsreisenden (TSP) als Graph

Problem: Finde für eine beliebige Menge von Städten mit gegebenen paarweisen Entfernungen die kürzeste Rundreise, die jede Stadt genau einmal besucht und zum Ausgangspunkt zurückkehrt.

Probleminstanz: 4 Städte A, B, C, D mit folgender Entfernungstabelle:

	A	B	C	D
A	–	10	15	20
B	10	–	35	25
C	15	35	–	30
D	20	25	30	–

Diese Tabelle ist eine **abstrahierte Darstellung** – sie enthält nur die für das Problem relevanten Informationen (Entfernungen), keine Straßennamen, keine Geographie etc.

📝 Übung 1 – Problem oder Probleminstanz?

Entscheidet für jede Aussage, ob es sich um ein **Problem** (allgemein) oder eine **Probleminstanz** (konkret) handelt, und begründet kurz:

1. "Packe möglichst wertvolle Gegenstände in einen Rucksack mit maximal 10 kg Tragkraft, wobei diese fünf Gegenstände zur Verfügung stehen: ..."
2. "Packe möglichst wertvolle Gegenstände in einen Rucksack mit begrenzter Tragkraft."
3. "Finde den kürzesten Weg von Saarbrücken nach Trier im gegebenen Straßennetz."
4. "Finde in einem beliebigen Graphen den kürzesten Weg zwischen zwei beliebigen Knoten."

(Eure Antworten hier)

📝 Übung 2 – Symmetrien erkennen

Betrachtet die Entfernungstabelle aus dem Beispiel oben.

1. Welche strukturelle Eigenschaft erkennt ihr an der Tabelle bezüglich `Entfernung(A,B)` und `Entfernung(B,A)`? Was bedeutet das für die Anzahl der Rundreisen, die ihr tatsächlich unterscheiden müsst?
2. Wie viele **verschiedene** Rundreisen (unter Berücksichtigung dieser Symmetrie) gibt es bei 4 Städten? Zählt systematisch.

(Eure Antworten hier)

🐍 Optionale Veranschaulichung mit Python

Die folgende Zelle listet alle möglichen Rundreisen für die obige Probleminstanz auf – nur zur Veranschaulichung, keine Pflichtaufgabe.

```python
from itertools import permutations

staedte = ["A", "B", "C", "D"]
start = staedte[0]
restliche = staedte[1:]

rundreisen = list(permutations(restliche))
print(f"Anzahl möglicher Rundreisen (ohne Symmetriebetrachtung): {len(rundreisen)}")
for reise in rundreisen:
    print(start, "->", " -> ".join(reise), "->", start)
```

```
Anzahl möglicher Rundreisen (ohne Symmetriebetrachtung): 6
A -> B -> C -> D -> A
A -> B -> D -> C -> A
A -> C -> B -> D -> A
A -> C -> D -> B -> A
A -> D -> B -> C -> A
A -> D -> C -> B -> A
```

🚀 Zusatzaufgabe

Formuliert selbst ein Problem und eine dazu passende Probleminstanz aus einem Kontext eurer Wahl (z. B. Stundenplanung, Einkaufstour, Gruppeneinteilung).