

## Methods of Artificial Intelligence

**Programming Task** 



## **Programming Task**



Problem: You have to implement a planning module for an intelligent warehouse system that

- has access to all items available in the warehouse and which PSUs carry which items
- given an order, decides which PSUs to pick and to transport to the human operator for packing
- you want to minimize the number of PSUs that have to be carried

#### Warehouse information,



Information about the items in the warehouse will be made available as a text file for you

- The first line contains all items separated by space
- Each of the following lines, represents the items stored by one PSU
- You will have to write (and test) a file reader

#### Example Warehouse File

```
i1 i2 i3 i4 i5 i6 i7 i8

i1 i4 i8
i2 i3 i4
i1 i5 i8
i6 i7
i2 i7 i8
```

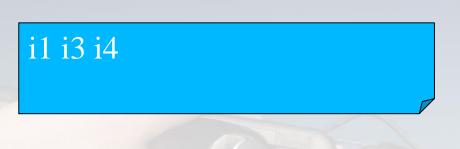
#### Order information



Information about orders will also be made available as a text file

• There is only one line containing all items in the order separated by space

# Example Order File



#### Graphical User Interface



The planning problem is an optimization problem that may remind you of a problem that you saw in the course. Implement a simple graphical user interface that allows the user to

- Read a text file with warehouse information
- Read an order
- Compute a selection of PSUs that satisfy the order and print the solution
- Select from different local search algorithms

### Solution Output



After having found a solution, print the following information

- Number of PSUs used
- For every PSU that was used, print also
  - identifier of the PSU
  - followed by items stored in the PSU

### Algorithms



The user should be able to select from the following Algorithms:

- Hill-Climbing
- First-Choice Hill-Climbing
- One of Random Restart or Parallel Hill-Climbing with configurable number of states
- Simulated Annealing
- Local Beam Search with configurable number of states

#### Other Requirements



After printing the solution, the user should be able to

- Rerun the algorithm (random initialization)
- Try another algorithm
- Replace the current order with a new one
- Replace the current warehouse configuration with a new one

without having to restart the application.

The application should also be intuitive and easy to use

## **Programming Languages**



We recommend that you use either

- Java or
- Python

If you want to use another language, talk to us first

#### User Interface



- Preferably GUI that shows options and results in a well organized window
- Text-based UI is also okay if it can be used easily and intuitively
- design user interface conceptually before implementing it and check with your tutors if your design is okay
- Recommended libraries
  - Java: Swing
  - Python: Tkinter or QT

#### Documentation



#### Code structure and documentation will influence your grade

- Every module should start with a comment that explains what the module is good for
- Use reasonable variable and function names
- Add comments where functionality is not self-explaining
- There should be a readme that explains the overall structure of the code (the GUI is supposed to be self-explaining) and the version numbers of all libraries that you used