



```
(define (domain warehouse)
  (:requirements :strips :typing :adl :fluents :durative-actions)
  (:types
    robot
    zone
    pallet)
  (:predicates
    (robot_available ?robot - robot)
    (robot_at ?robot - robot ?zone - zone)
    (pallet_at ?pallet - pallet ?zone - zone)
    (pallet_not_moved ?pallet - pallet)
    (is_unload_zone ?zone - zone)
    (is_shelf_zone ?zone - zone)
    (is_recharge_zone ?zone - zone))
  (:functions
    (distance ?from - zone ?to - zone)
    (speed ?robot - robot)
    (battery_level ?robot - robot)))
```

```
(define (problem robot_warehouse)
  (:domain warehouse)
  (:objects
    tars - robot
    charging_station unload_zone shelf_1
    shelf_2 shelf_3 shelf_4 - zone)
  (:init
    (robot_at tars charging_station)
    (robot_available tars)
    (is_recharge_zone charging_station)
    (is_unload_zone unload_zone)
    (is_shelf_zone shelf_1)
    (is_shelf_zone shelf_2)
    (is_shelf_zone shelf_3)
    (is_shelf_zone shelf_4)
    (= (battery_level tars) 100)
    (= (speed tars) 0.26)
    (= (distance charging_station unload_zone) computed_distance)
    (= (distance unload_zone charging_station) computed_distance)
    (= (distance charging_station shelf_1) computed_distance)
    (= (distance shelf_1 charging_station) computed_distance)
    (= (distance charging_station shelf_2) computed_distance)
    (= (distance shelf_2 charging_station) computed_distance)
    (= (distance charging_station shelf_3) computed_distance)
    (= (distance shelf_3 charging_station) computed_distance)
    (= (distance charging_station shelf_4) computed_distance)
    (= (distance shelf_4 charging_station) computed_distance)
    ...
    ...
    ...)
```