A* Search Report

Group 14
Michael King, Matthew Reilly, Kerr Brydon

We have created a program to find a path to sort the packages between the warehouses and the truck using an A* search implemented in java. The warehouses and the truck were coded as nodes stored in an arraylist. Each node is made up of an arraylist of child nodes, an arraylist of costs, a parent node, the location, the number of each type of package, a heuristic value, a path cost value, and the sum of the last two values. The node also has functions to return the number of packages, the heuristic value, and the sum value.

Two heuristic functions have been implemented. The first heuristic function works by totalling the number of packages in the wrong location. This value is used in conjunction with the path cost to guide the robot. The second heuristic function finds the type of package that appears the most outside its intended location. Both heuristics will find the path taking the least number of moves although they may find slightly differing paths that take the same amount of steps.

The A* search works by creating child nodes for the current node and giving these child nodes the values corresponding to the two locations the robot is not currently at e.g. if the robot is at the truck, the child nodes will be set to be warehouse A and warehouse B. The program will then check if there are any packages to be moved from the robot's current location to either of the child nodes. The next step involves calculating the heuristic values for both children. Depending on which returns a lower value one node will be iterated through while the other will be added to a list of missed nodes to be checked once the other node's paths have been exhausted. If a path is found that reaches the goal state of all packages being placed in the correct location, then true is returned along with the path and it's cost. This path is then looped through with each step printed to the screen.