CZ3005: Artificial Intelligence

LAB ASSIGNMENT 1: Finding a Shortest Path with An Energy Budget

2021/2022 SEMESTER 2

Group - ILoveMDP2

Li Zhaoyuan (U2020129C)

Lim Rui An, Ryan (U2022692K)

Task 1: Any search with no energy constraint

- Algorithm Implemented: **Dijkstra Greedy Search** [1]
    - Since this task is equivalent to a shortest path problem, we decided that a greedy approach will be the best answer as the algorithm will prioritise on the shortest path.
- Approach Used For Dijkstra Greedy Algorithm:
    - Shortest distance, previous nodes and visited status are all stored in separated arrays.
    - Nodes that will be chosen to be explored will be selected from a priority queue sorted by their respective distance apart.
        - Chosen nodes will be compared with it's adjacent nodes and if the path cost to the current adjacent nodes is lower than previous found value, the shortest distance and the previous nodes array will be updated.
    - This process will continue till there are no longer any nodes left in the priority queue to be explored and a loop is used to traverse the previous nodes array to print the shortest path. The shortest distance can also be found in the shortest distance array.
    - Since there is no energy constraint, there is no answer for total energy cost.
- Output:

  Shortest path:

  1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->988->979->980->969->977->989->990->991->2369->2366->2340->2338->2339->2333->2334->2329->2029->2027->2019->2022->2000->1996->1997->1993->1992->1989->1984->2001->1900->1875->1874->1965->1963->1964->1923->1944->1945->1938->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50

  Shortest distance: 148648.63722140007

  Total energy cost: nil

```
Task 1 Greedy Path From  1 to 50
Shortest path: 1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->988->979->980->969->977->989->990->991->2369->2366->2340->2338->2339->2333->2334->2329->2029->2027->2019->2022->2000->1996->1997->1993->1992->1989->1984->2001->1900->1875->1874->1965->1963->1964->1923->1944->1945->1938->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50
Shortest distance: 148648.63722140007
Total energy cost: nil
```

Task 2: Any uninformed search algorithm

- Algorithm Implemented: **Uniform Cost Search** [2]
    - UCS is used for this problem to find the lowest cost path from the start node to the goal node as all edges in the graph may have differing costs
    - UCS is similar to Breadth First Search (BFS) but makes use of a priority queue to order the nodes to be visited next
- Approach Used For Uniform Cost Search:
    - Initialise a PriorityQueue with the starting node to represent the Frontier to be explored and dictionaries to store Explored nodes, DistanceCost & EnergyCost
    - Iteratively explore nodes in the Frontier as long as they are within the energy budget, recording the "Distance, Energy, Node" for neighbour nodes to this current node and inserting them into the Frontier for future exploration
    - Repeat this process of exploring the frontier until the Goal node is extracted from the frontier to be explored
    - Similar to Task 1, the explored nodes are traversed from the Goal node to output the shortest path and identify the distance and energy.
- Output:

  Shortest Path:
  1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->986->979->980->969->977->989->990->991->2465->2466->2384->2382->2385->2379->2380->2445->2444->2405->2406->2398->2395->2397->2142->2141->2125->2126->2082->2080->2071->1979->1975->1967->1966->1974->1973->1971->1970->1948->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50

  Shortest Distance: 150335.55441905273

  Total Energy Cost: 259087

```
> TASK 2: Results of Uniform Cost Search
Shortest Path: 1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->986->979->980->969->977->989->990->991->2465->2466->2384->2382->2385->2379->2380->2445->2444->2405->2406->2398->2395->2397->2142->2141->2125->2126->2082->2080->2071->1979->1975->1967->1966->1974->1973->1971->1970->1948->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50
Shortest Distance:  150335.55441905273
Total Energy Cost:  259087
```

Task 3: A* search algorithm

- Algorithm Implemented: **A* Search** [2]
    - Implemented A* Search by modifying Task 2's Uniform Cost Search.
- Approach Used For A* Search:
    - Added a heuristic function that is used to calculate the coordinate distance from the current point to the next point.
    - The heuristic value calculated from the function, F(n) = G(n) + H(n), where the function G represents the path cost and H represents the coordinate distance, the heuristic, F(n) is used to determine the priority of the nodes to be explored next located in the Frontier
- Output:
    - Shortest Path:
      1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->988->979->980->969->977->989->990->991->2465->2466->2384->2382->2385->2379->2380->2445->2444->2405->2406->2398->2395->2397->2142->2141->2125->2126->2082->2080->2071->1979->1975->1967->1966->1974->1973->1971->1970->1948->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50
    - Shortest Distance: 150335.55441905273
    - Total Energy Cost: 259087

```
> TASK 3: Results of A* Search Algorithm
Shortest Path: 1->1363->1358->1357->1356->1276->1273->1277->1269->1267->1268->1284->1283->1282->1255->1253->1260->1259->1249->1246->963->964->962->1002->952->1000->998->994->995->996->987->988->979->980->969->977->989->990->991->2465->2466->2384->2382->2385->2379->2380->2445->2444->2405->2406->2398->2395->2397->2142->2141->2125->2126->2082->2080->2071->1979->1975->1967->1966->1974->1973->1971->1970->1948->1937->1939->1935->1931->1934->1673->1675->1674->1837->1671->1828->1825->1817->1815->1634->1814->1813->1632->1631->1742->1741->1740->1739->1591->1689->1585->1584->1688->1579->1679->1677->104->5680->5418->5431->5425->5424->5422->5413->5412->5411->66->5392->5391->5388->5291->5278->5289->5290->5283->5284->5280->50
Shortest Distance:  150335.55441905273
Total Energy Cost:  259087
```

Contribution

Name	Work Done
Li Zhaoyuan (U2020129C)	Task 1, Task 3, Report
Lim Rui An, Ryan (U2022692K)	Task 2, Task 3, Report

Conclusion

- We learnt how to write uninformed searches with and without having an additional constraint.
- We learnt about how to modify the search algorithm when dealing with additional constraints.
    - Dealing with constraint was not just another additional If statement check, but rather we have to consider all possibilities of each dist, energy pair for each node to determine the shortest path with the best energy cost.
- We got to experience working with a big dataset with more than thousands of different nodes.
    - Due to the large dataset, visualisation was an issue and we managed to do some backtracking to check if our path was correct.
- A practice with writing Dijkstra Greedy Algorithm and Uniform Cost Search.
- There are multiple ways we can choose to calculate heuristic for A* Search.

References

- [1] - F. Lin , “CX2101 Algorithm Design and Analysis, Week 4-2 and 5-1 Greedy Algorithms-S1,” *NTULearn*. [Online]. Available: https://ntulearn.ntu.edu.sg/. [Accessed: 13-Mar-2022].
- [2] - H. Zhang, “CZ3005 Artificial Intelligence, Module 2 Intelligent Agents & Search,” *NTULearn*. [Online]. Available: https://ntulearn.ntu.edu.sg/. [Accessed: 13-Mar-2022].