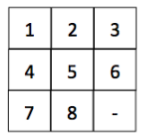
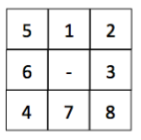
## Note:

Write a group report that summarises the answers you obtain and submit as a PDF file to the Moodle web-site.

Answers…………………………………..

Note: Add these two images beginning of the report showing start & end state

Start End

1. Answers for first Question

* Searching Algorithm: Breadth-First Search

Heuristic Function: Not Applicable

1.1Depth **=** 10

1.2 Explored Number of States = 421

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) > **6**(d10)

* Searching Algorithm: Depth-First Search

Heuristic Function: Not Applicable

1.1Depth **=** 1000 > Can’t identify exactly, Browser crash before reach the goal

state

1.2 Explored Number of States = 1000 > Can’t identify exactly, Browser crash

before reach the goal state

1.3 Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **……..**

* Searching Algorithm: Iterative Deepening Search

Heuristic Function: Not Applicable

1.1Depth **=** 10

1.2 Explored Number of States = 454

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) > **6**(d10)

* Searching Algorithm: A\* Search
  + Heuristic Function: Euclidean Distance

1.1Depth **=** 10

1.2 Explored Number of States = 19

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

* + Heuristic Function: Manhattan Distance (City-Block distance)

1.1Depth **=** 10

1.2 Explored Number of States = 11

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

* + Heuristic Function: Tiles Out-of-place

1.1Depth **=** 10

1.2 Explored Number of States = 31

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

* Searching Algorithm: Greedy Search
  + Heuristic Function: Euclidean Distance

1.1Depth **=** 10

1.2 Explored Number of States = 11

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

* + Heuristic Function: Manhattan Distance (City-Block distance)

1.1Depth **=** 10

1.2 Explored Number of States = 11

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

* + Heuristic Function: Tiles Out-of-place

1.1Depth **=** 10

1.2 Explored Number of States = 12

1.3 Path = [ ***d***x = in ***depth*** x ]

**8**(d1) > **7**(d2) > **4**(d3) > **5**(d4) > **6**(d5) > **3**(d6) > **2**(d7) > **1**(d8) > **5**(d9) >

**6**(d10)

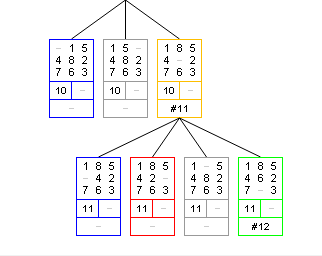
1. Yes.

Optimal Solution find on Greedy Search Algorithm. When using Greedy search both

Euclidean Distance & Manhattan Distance(City-Block distance) Heuristic Functions reached

the goal state on 11 number of explored states & only Tiles Out-of-place Heuristic Function took 12 number of explored states, although it’s better than every other Uninformed and Informed Search Algorithms.

1. On Depth-First Search path of cost 12



Therefore Propose End State is



Other algorithms perform on this end state as below:

* Searching Algorithm: Breadth-First Search

Heuristic Function: Not Applicable

Depth **=** 11

Explored Number of States = 1102

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

To reach this end state, this algorithm take more time than Depth-First Search Algorithm.

* Searching Algorithm: Iterative Deepening Search

Heuristic Function: Not Applicable

Depth **=** 11

Maximum depth: 12

Explored Number of States = 12

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

To reach this end state, this algorithm take more time than Depth-First Search Algorithm.

* Searching Algorithm: A\* Search
* Heuristic Function: Euclidean Distance

Depth **=** 11

Explored Number of States = 46

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

* + Heuristic Function: Manhattan Distance (City-Block distance)

Depth **=** 11

Explored Number of States = 17

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

* + Heuristic Function: Tiles Out-of-place

Depth **=** 11

Explored Number of States = 63

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

* Searching Algorithm: Greedy Search
  + Heuristic Function: Euclidean Distance

Depth **=** 11

Explored Number of States = 20

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)

* + Heuristic Function: Manhattan Distance (City-Block distance)

Depth **=** 23

Explored Number of States = 48

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **5**(d2) > **8**(d3) > **6**(d4) > **5**(d5) > **3**(d6) > **….** > **5**(d21) > **2**(d22) > **3**(d23)

* + Heuristic Function: Tiles Out-of-place

Depth **=** 11

Explored Number of States = 228

Path = [ ***d***x = in ***depth*** x ]

**6**(d1) > **3**(d2) > **2**(d3) > **5**(d4) > **8**(d5) > **6**(d6) > **3**(d7) > **2**(d8) > **5**(d9) > **8**(d10) >

**6**(d11)