

Lab2

Name: Zeng Yuhang

ID: 222320008

Variant: 5 (Nanjing to Dalian)



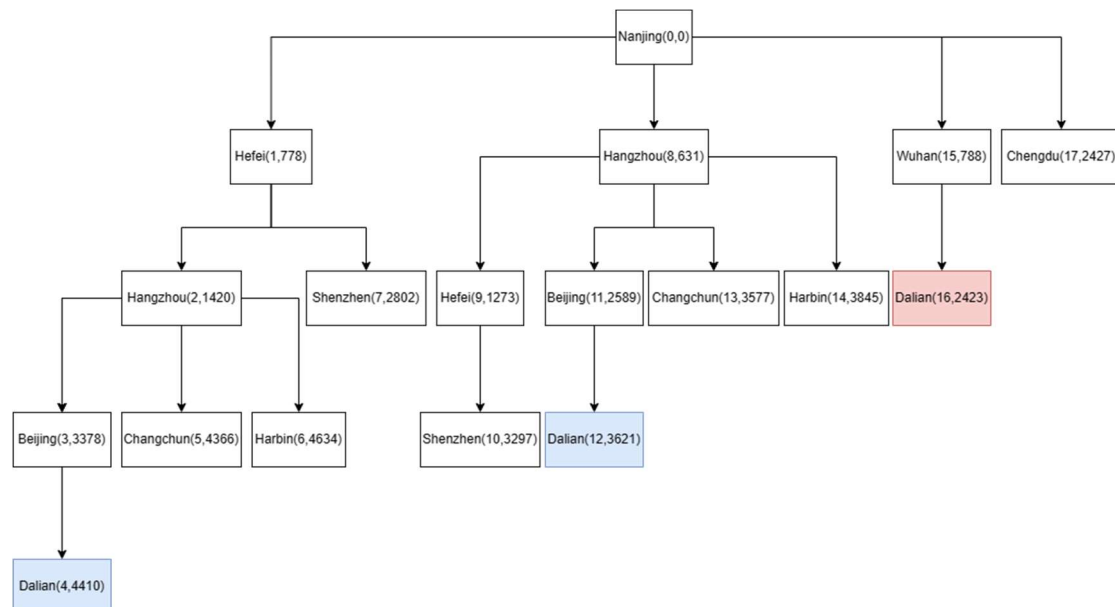
Task statement

The main purpose of this lab work is trying to understand 5 kinds of search methods and compare the differences among them. The search methods include Depth first search, Width first search, Bidirectional search, Greedy search and A* search. In order to help us understand, we need to draw a search tree for each method and describe the characteristics of each search method.

Graph of the search tree

I assume that the neighbors of each node are arranged by distance. The two numbers in () means this node will be visited at which step and current distance respectively. Blue means changing the best route and distance after visiting this node. Red means the final result. The conditions of terminating the search contain: reach Dalian, current node has no more path, current distance is larger than best distance.

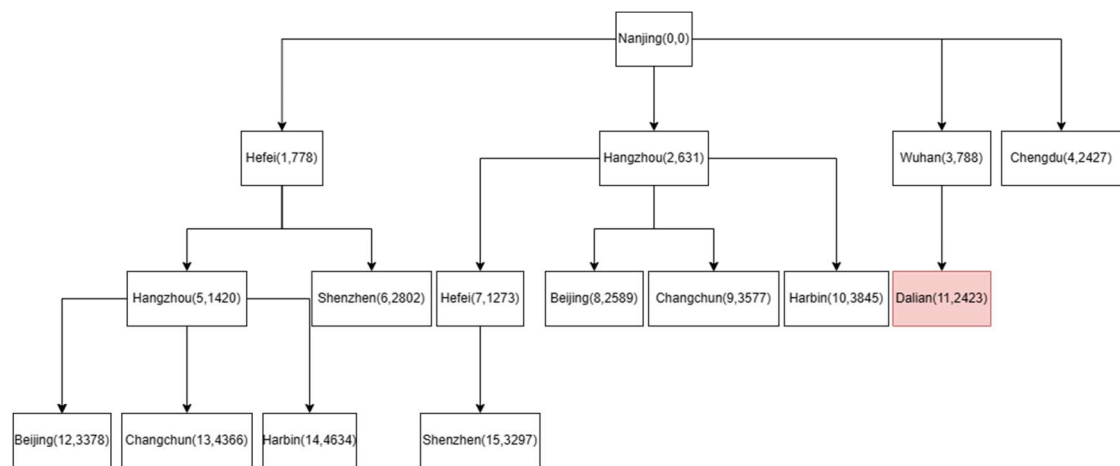
1. Depth first search



route = [Nanjing, Wuhan, Dalian], distance = 2423

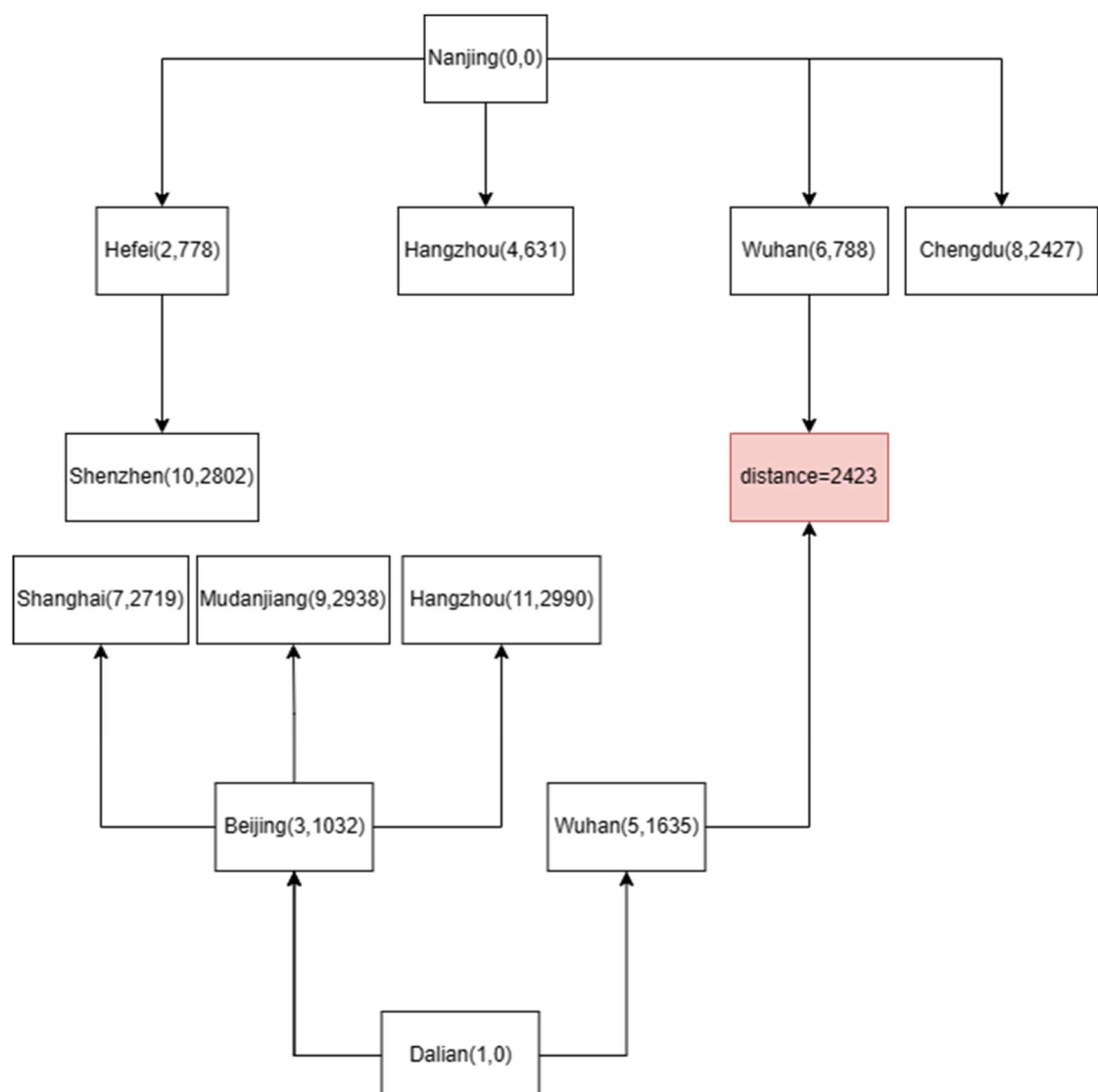
We can easily find some redundant path in finding, so there must be some solutions to optimize DFS, such as store the best distance between all visited nodes.

2. Width first search



route = [Nanjing, Wuhan, Dalian], distance = 2423

3. Bidirectional search



route = [Nanjing, Wuhan, Dalian], distance = 2423

4. Greedy search

Hefei – Dalian: 960

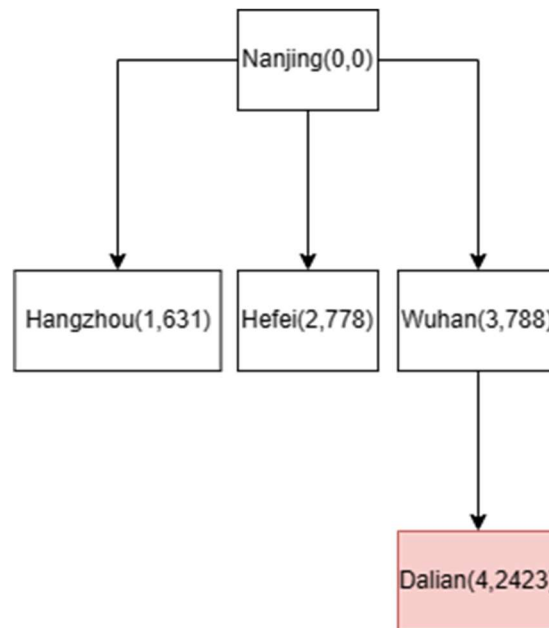
Hangzhou – Dalian: 1005

Wuhan – Dalian: 1188
Chengdu – Dalian: 1781
Shenzhen – Dalian: 2004
Beijing – Dalian: 443
Changchun – Dalian: 694
Harbin – Dalian: 882



route = [Nanjing, Hefei, Hangzhou, Beijing, Dalian], distance = 3772

5. A* search



route = [Nanjing, Wuhan, Dalian], distance = 2423

Reached characteristics of each search method

1. Depth first search

It starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking. Characteristics of DFS include that it is recursive and uses a stack to keep track of visited nodes.

2. Width first search

It starts at the tree root (or some arbitrary node of a graph) and explores all the neighbor nodes at the present depth before moving on to the nodes at the next depth level. This algorithm is implemented using a queue data structure to store nodes and their neighbors.

3. Bidirectional search

It runs two simultaneous searches: one forward from the initial state and the other backward from the goal. The search terminates when two searches meet in the middle.

4. Greedy search

It is called greedy because it makes the best choice at each step without considering the future consequences. Greedy search algorithms are often used in optimization problems where the goal is to find the best solution among a set of possible solutions. However, greedy algorithms do not always guarantee an optimal solution.

5. A* search

It uses a heuristic function to estimate the cost of the cheapest path from the start node to the goal node. The heuristic function is an estimate of the remaining distance to the goal node. The A* algorithm combines this heuristic with a cost function that gives the cost of the cheapest path from the start node to any node n . The cost function is usually denoted $g(n)$. The total estimated cost of a path through node n to the goal node is $f(n) = g(n) + h(n)$, where $h(n)$ is the heuristic estimate of the remaining distance to the goal.

node.

Summary

The four of five search methods mentioned return the right answer. Depth first search, Width first search and Bidirectional search can guarantee the right answer by using more times. Bidirectional search is relatively quicker. Greedy search and A* search can't guarantee the right answer. However, in the condition of my variant, A* get the truth.