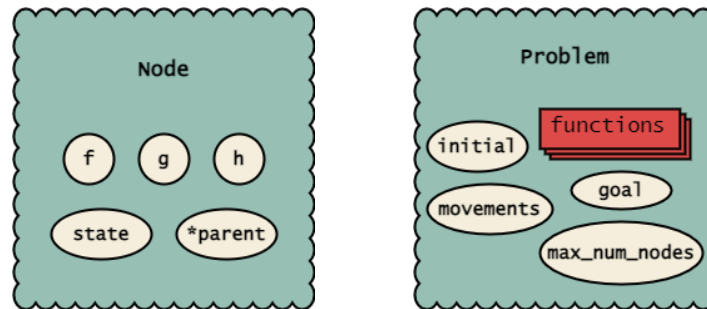


N-Puzzle

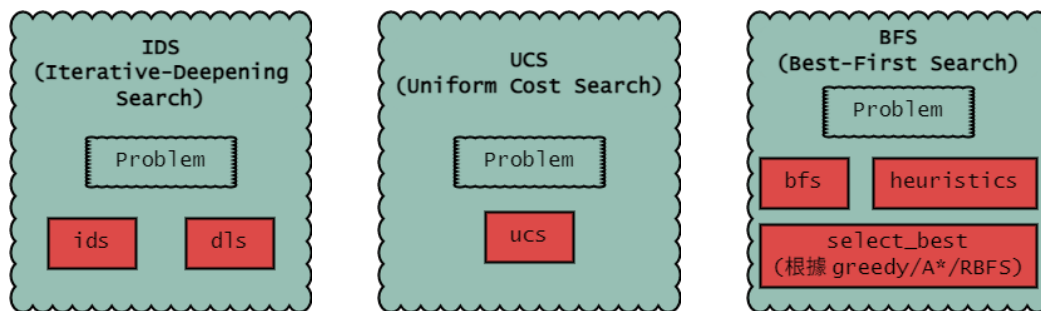
410885010 劉嘉蕎

A. Class : Node 、 Problem



- 、 Node 說明：在建構函式內就已計算好 g，而 f、h 只有在 Best-First Search 才會用到，所以 f、g 的計算由類別 BFS 負責(後面會說明)。
- 、 Problem 的函式
 - a. `void pop_i(vector<Node*>& vec, int i)` //刪除指標 vector 中第 i 元素
 - b. `bool isgoal(Node* node)` //判斷節點是否為目標
 - c. `bool iscycle(Node* node)` //判斷節點是否進入循環
 - d. `bool inside(vector<Node*> &reached, string state)` //判斷指標 vector 裡面是否有指向狀態為 state 的節點指標
 - e. `void expand(vector<Node*> &frontier, Node* parent)` //為 parent 節點建立子節點並加入 frontier
 - f. `vector<string> action(string s)` //被上述函式 expand()呼叫，返回所有子節點應有的狀態
 - g. `void make_path(Node* last_node)` //生成完整路徑，同時呼叫 print_path()，並且印出 movements 和 max_num_nodes 的值
 - h. `void print_path(string result)` //被上述函式 make_path()呼叫，印出完整路徑

C. 繼承 Problem 的類別：IDS、UCS、BFS



1、IDS 的函式

- a. `void ids()` // 跟上課講義一樣
- b. `string dls(unsigned l)` // 跟上課講義一樣

、UCS 的函式

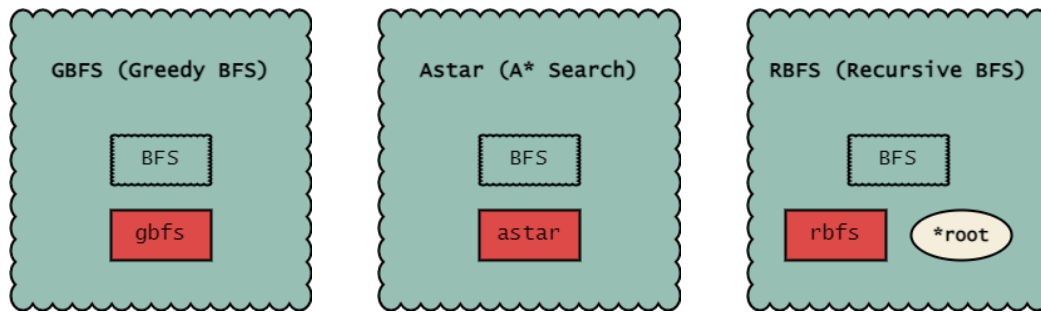
- a. `void ucs()` // 跟上課講義一樣

、BFS 的函式

跟上課講義一樣

- b. `int select_best(vector<Node*> &frontier, string mode)` // 根據模式 mode，選出 frontier 中的最佳節點，回傳指向該節點的指標在 vector 中的索引值(mode = “greedy”, “astar”, “best_f”, “second_best_f”，後兩種模式使用於 Recursive BFS)
- c. `void heuristics(Node* node)` // 計算節點的 h 與 f

D. 繼承 BFS(Best-First Search)的類別：GBFS、Astar、RBFS



1、GBFS 的函式

- a. `void gbfs() //呼叫 bfs()，mode = “greedy”`

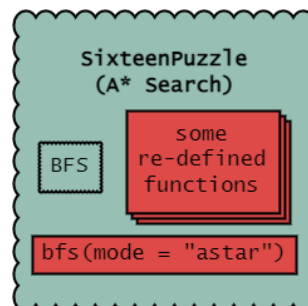
2、Astar 的函式

- a. `void astar() //呼叫 bfs()，mode = “astar”`

3、RBFS 的函式

- a. `pair<string, unsigned> rbfs(Node* node, unsigned f_limit) //跟上課講義一樣`

E. 加分題：使用 A* Search，因此同樣繼承 BFS，有重寫幾個會被影響的函式



1、被重寫的函式

- a. `void heuristics(Node* node) //盤面改變，f、h 的計算方法也要改變`
- b. `void bfs(string mode) //有呼叫上述 heuristics()，所以需重新定義`
- c. `void print_path(string result) //盤面改變，印出盤面的函式也要改變`
- d. `void make_path(Node* last_node) //有呼叫上述 print_path()，所以需重新定義`

- e. `vector<string> action(string s)` //盤面改變，回傳子節點狀態的函式也要改變
- f. `void expand(vector <Node*> &frontier, Node* parent)` //有呼叫上述 `action()`，所以需重新定義

F. 操作說明

1、 在 `main()` 先修改 `initial` 值再編譯、執行。

2、 `bonus_initial` 數字與對應的字元：

10	A
11	B
12	C
13	D
14	E
15	F