



الجمهورية العربية السورية

جامعة دمشق

كلية الهندسة المعلوماتية

مشروع مادة خوارزميات بحث ذكية

المسار الأقصر

إعداد الطلاب:

أيهم الرفاعي – عبد الرحمن حجازي – وائل عرابي

براء الدوماني

إشراف:

معيدين العملي

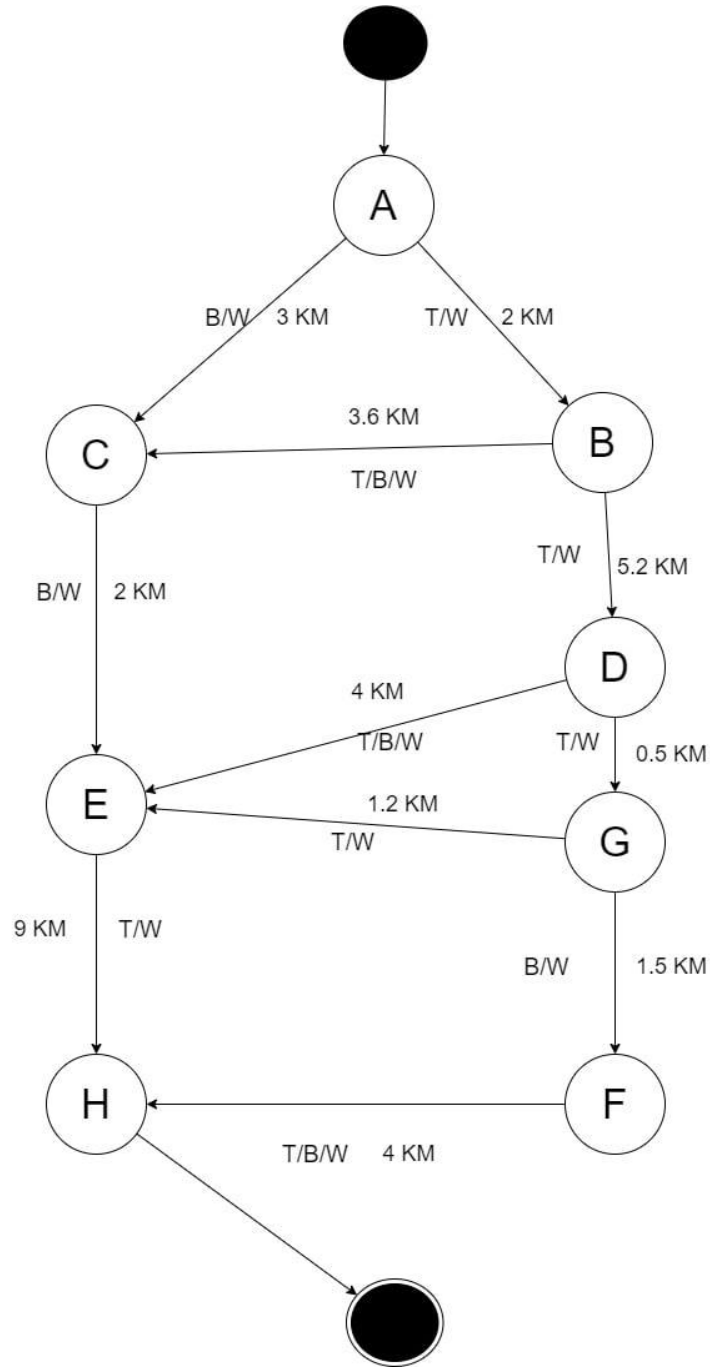
التاريخ : 5 / 1 / 2023

الأفكار الرئيسية :

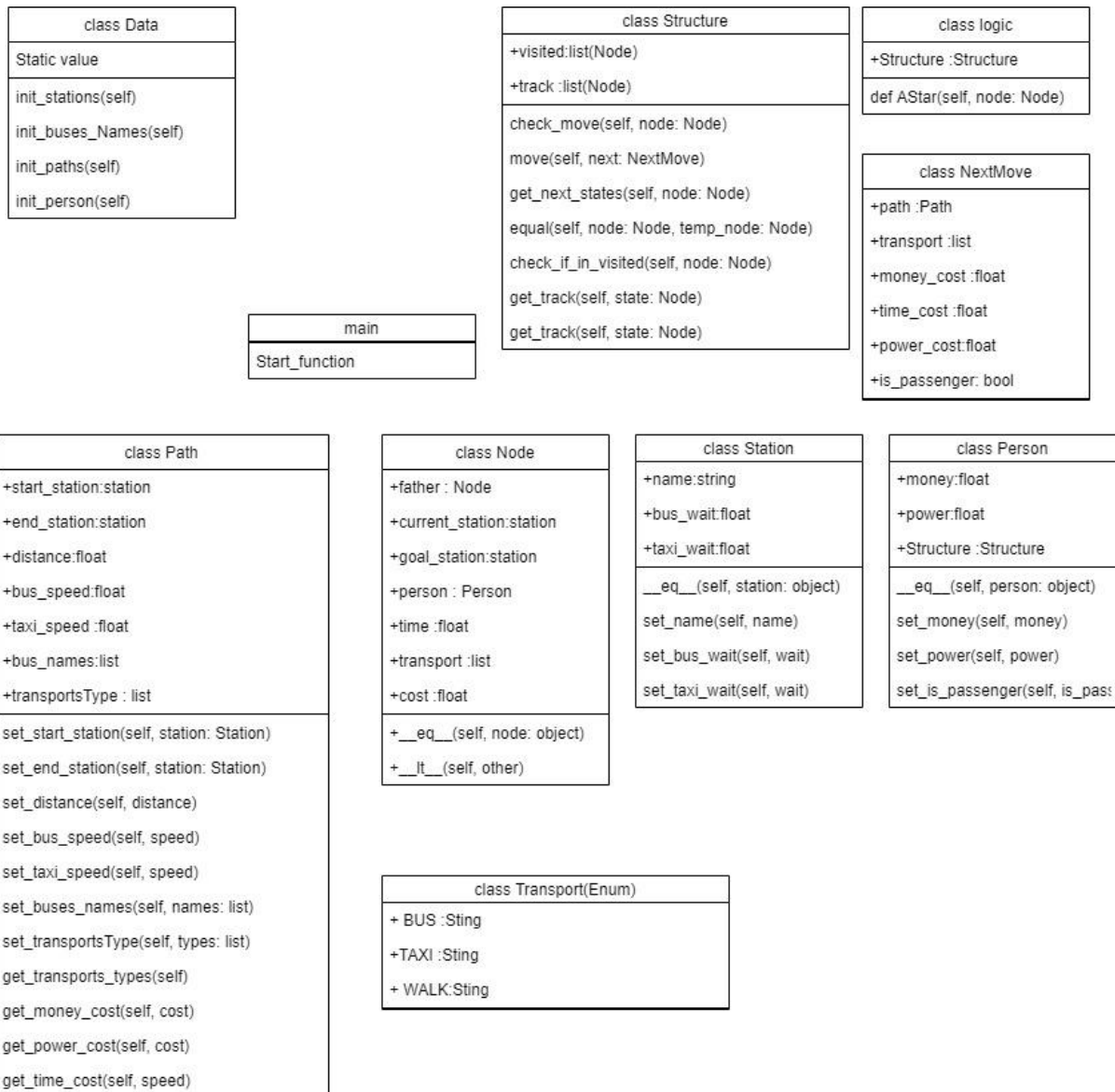
1. الهدف هو الوصول إلى المنزل مع مراعاة الكلفة المرادة
2. يجب تمثيل الكلاسات التالية : (تم تمثيلها في لغة python)

1. Station : يمثل المحطات بين الطرق وأوقات الانتظار لكل وسيلة مواصلات
2. Path : يمثل الطريق بين محطتين محطة بداية ونهاية والمسافة بينهما ، بالإضافة الى سرعة المواصلات في هذا الطريق وأسماء خطوط الباصات اذا كانت متوفرة
3. Transport : هو كلاس يمثل وسائل النقل المتاحة من النمط enum
4. Person : يمثل الشخص الذي يستخدم الخوارزمية وتعريف لقيمه : متبقي ما معه من مال وطاقة
5. Node : يمثل المحطة الحالية والمحطة المرادة الطرق المتاحة بالمحطة الحالية ومعلومات الشخص و الوقت المستغرق الى وصله لهذه العقدة
6. Data : يمثل القيم المدخلة التي يتم استخدامها في اللعبة، تم ادخالها من خلال توابع init للكلاسات المعتمد عليها بحل المسألة مثل : Station , Path , Person
7. Logic : يحوي الخوارزميات التي يتم العمل عليها في اللعبة وهي AStar و BFS
8. Structure : يمثل بنية اللعبة وداخله قمنا بتعرف التوابع اللازمة لتحريك العقد واختبار الحالات
9. NextMove : يمثل طريقة للمرور من عقدة الى اخرى أي أنه يعيد الطريق واسلوب المرور مع الكلف لكل حركة

تم اعتماد هذا ال Graph كاختبار للعبة التي تم العمل عليها :



:Class Diagram



1. خوارزمية الـ BFS:

```
1  =====
2  Node : A
3  Time : 0:00
4  Money: 10000
5  power: 100
6  Transport : ['WALK', 'walk']
7  =====
8
9  =====
10 Node : C
11 Time : 0:32:43.636
12 Money: 10000
13 power: 70
14 Transport : ['WALK', 'walk']
15 =====
16
17 =====
18 Node : E
19 Time : 0:42:32.727
20 Money: 10000.0
21 power: 61.0
22 Transport : ['WALK', 'walk']
23 =====
24
25 =====
26 Node : H
27 Time : 1:30:05.454
28 Money: 1000.0
29 power: 106.0
30 Transport : ['TAXI', 'taxi']
31 =====
32
33 =====
34 Time Excution : 0:00:00.052
35 Visited node : 73
36 Track node : 4
37 =====
```

2. خوارزمية الـ AStar:

اقل كلفة :

```
1  =====
2  Node : A
3  Time : 0:00
4  Money: 10000
5  power: 100
6  Transport : ['WALK', 'walk']
7  =====
8
9  =====
10 Node : B
11 Time : 0:21:49.090
12 Money: 10000
13 power: 80
14 Transport : ['WALK', 'walk']
15 =====
16
17 =====
18 Node : D
19 Time : 0:48:38.181
20 Money: 4800.0
21 power: 106.0
22 Transport : ['TAXI', 'taxi']
23 =====
24
25 =====
26 Node : G
27 Time : 0:54:05.454
28 Money: 4800.0
29 power: 101.0
30 Transport : ['WALK', 'walk']
31 =====
32
33 =====
34 Node : F
35 Time : 1:10:27.272
36 Money: 4800.0
37 power: 86.0
38 Transport : ['WALK', 'walk']
39 =====
40
41 =====
42 Node : H
43 Time : 1:54:05.454
44 Money: 4800.0
45 power: 46.0
46 Transport : ['WALK', 'walk']
47 =====
48
49 =====
50 Time Excution : 0:00:00.015
51 Visited node : 40
52 Track node : 6
53 =====
```

```

1  =====
2  Node : A
3  Time : 0:00
4  Money: 10000
5  power: 100
6  Transport : ['WALK', 'walk']
7  =====
8
9  =====
10 Node : B
11 Time : 0:05
12 Money: 8000
13 power: 110
14 Transport : ['TAXI', 'taxi']
15 =====
16
17 =====
18 Node : D
19 Time : 0:10
20 Money: 2800.0
21 power: 136.0
22 Transport : ['TAXI', 'taxi']
23 =====
24
25 =====
26 Node : G
27 Time : 0:15:27.272
28 Money: 2800.0
29 power: 131.0
30 Transport : ['WALK', 'walk']
31 =====
32
33 =====
34 Node : F
35 Time : 0:21:15.272
36 Money: 2400.0
37 power: 123.5
38 Transport : ['BUS', 'GF']
39 =====
40
41 =====
42 Node : H
43 Time : 0:29:03.272
44 Money: 2000.0
45 power: 103.5
46 Transport : ['BUS', 'FH']
47 =====
48
49 =====
50 Time Excution : 0:00:00.031
51 Visited node : 58
52 Track node : 6
53 =====

```

```

1  =====
2  Node : A
3  Time : 0:00
4  Money: 10000
5  power: 100
6  Transport : ['WALK', 'walk']
7  =====
8
9  =====
10 Node : C
11 Time : 0:32:43.636
12 Money: 10000
13 power: 70
14 Transport : ['WALK', 'walk']
15 =====
16
17 =====
18 Node : E
19 Time : 0:42:32.727
20 Money: 10000.0
21 power: 61.0
22 Transport : ['WALK', 'walk']
23 =====
24
25 =====
26 Node : H
27 Time : 1:30:05.454
28 Money: 1000.0
29 power: 106.0
30 Transport : ['TAXI', 'taxi']
31 =====
32
33 =====
34 Time Excutio : 0:00:00.001
35 Visited node : 9
36 Track node : 4
37 =====

```


تقسيم العمل:

أيهم الرفاعي	تكوين البنية كاملة كلاس الادخال
وائل عرابي	تكوين البنية كاملة تنسيق عملية الطباعة والقيم
عبد الرحمن حجازي	خوارزمية الـ AStar وبنية كلاس Structure
براء الدوماني	خوارزمية الـ AStar وبنية كلاس Structure

شرح التوابع الرئيسية :

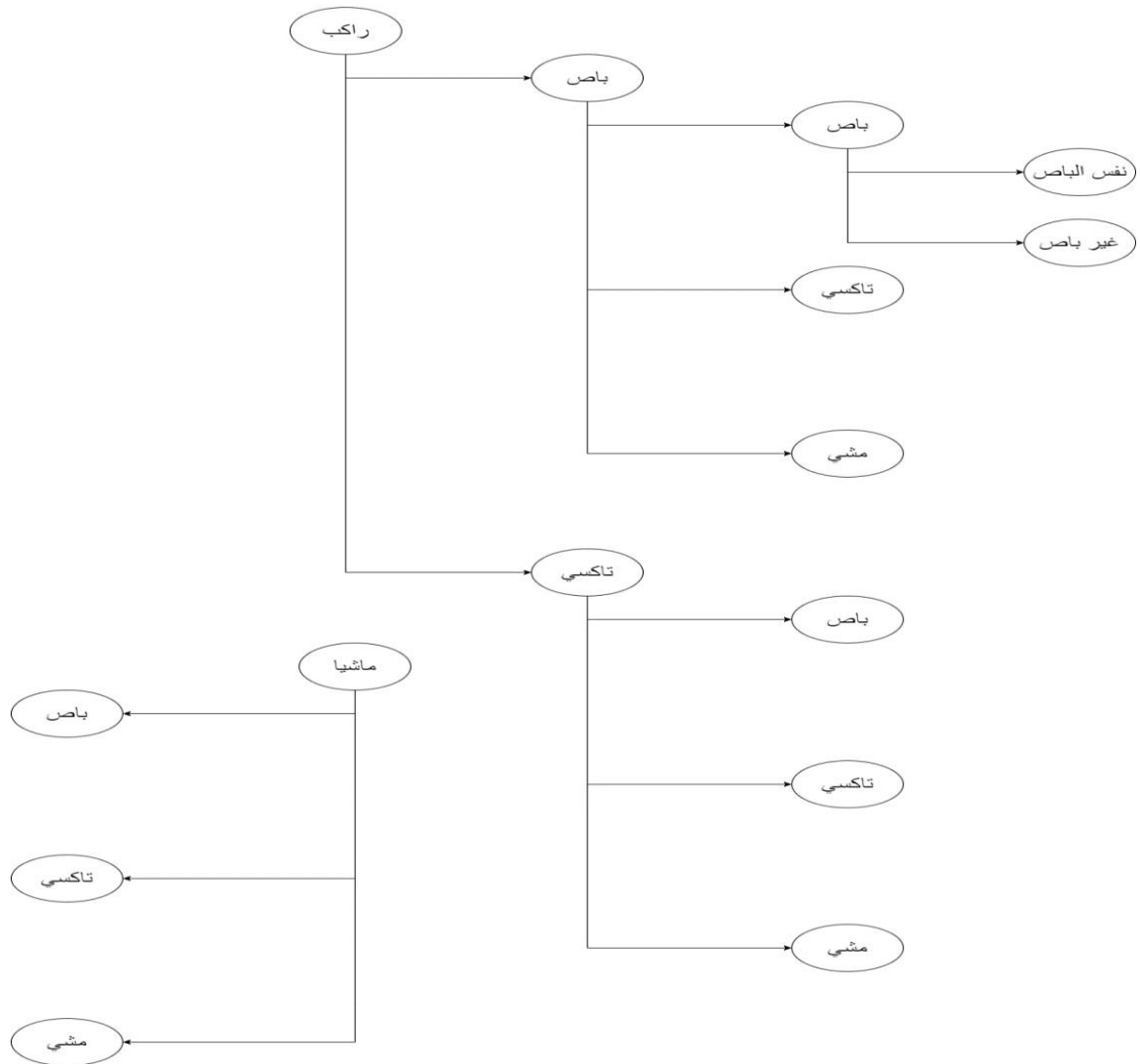
1. Check_move :

تابع يقوم باخذ عقدة ويعيد جميع أساليب المرور للمحطة التالية
نقوم بعمل حلقة على جميع المسارات لدينا مع شرط اذا كان اسم نقطة البداية في
الطريق تساوي النقطة الحالية

2. Move :

تابع يقوم بعمل نقل من المحطة الحالية للمحطة التالية مع الاخذ بعين الاعتبار
وسيلة المواصلات المأخوذة بالعقدة سابقا
ووسيلة المواصلات التي ستأخذ بالعقدة الحالية
يأخذ التابع اوبجكت من NextMove وهو الطريق المتاح مع أسلوب المرور

move



3. `:Get_next_states`
تابع يعيد جميع أبناء العقدة الممرة للتابع
4. `:Is_final_state`
يقوم باخبار اذا كانت هذه العقدة هي الهدف (المنزل) ام لا
5. `: get_track`
يقوم بإعادة ليست المسارات التي مر بها الشخص
6. `print_soluation`
يقوم بطباعة حلول الخوارزمية
7. `Equal` يقوم بتأكد من تساوي عقدتين

الخوارزمية Astar:

```
1 def AStar(self):
2     now = time.time()
3     queue = Queue.PriorityQueue()
4     queue.put(self.structure.node)
5     self.structure.visited.append(self.structure.node)
6     while queue.queue:
7         temp_node = queue.queue.pop(0)
8         if self.structure.is_final_state(temp_node):
9             self.structure.get_track(temp_node)
10            break
11        for state in self.structure.get_next_states(temp_node):
12            if (not self.structure.check_if_in_visited(state)):
13                queue.put(state)
14                self.structure.visited.append(state)
15
16        self.structure.print_soluation(now)
17
```