

الجمهورية العربية السورية جامعة دمشق كلية الهندسة المعلوماتية

# مشروع مادة خوارزميات بحث ذكية المسار الأقصير

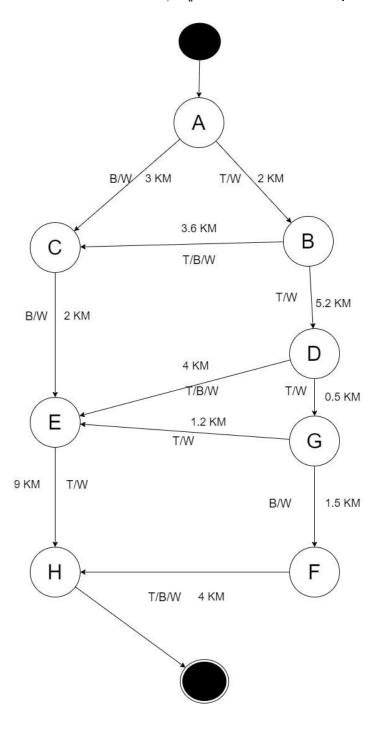
إعداد الطلاب:
أيهم الرفاعي — عبد الرحمن حجازي — وائل عرابي براء الدوماني الشراف:
اشراف:
معيدين العملى

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 ، بالاضافة الى سرعة الموصلات 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 الخوار زمية 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

class Data

Static value

init\_stations(self)

init\_buses\_Names(self)

init\_paths(self)

init\_person(self)

class Structure

+visited:list(Node)

+track :list(Node)

check\_move(self, node: Node)

move(self, next: NextMove)

get\_next\_states(self, node: Node)

equal(self, node: Node, temp\_node: Node)

check\_if\_in\_visited(self, node: Node)

get\_track(self, state: Node)

get\_track(self, state: Node)

class logic

+Structure :Structure

def AStar(self, node: Node)

class NextMove

+path :Path

+transport :list

+money\_cost :float

+time\_cost :float

+power\_cost:float

+is\_passenger: bool

class Path

+start\_station:station

+end\_station:station

+distance:float

+bus\_speed:float

+taxi\_speed :float

+bus\_names:list

+transportsType : list

set\_start\_station(self, station: Station)

set\_end\_station(self, station: Station)

set\_distance(self, distance)

set\_bus\_speed(self, speed)

set\_taxi\_speed(self, speed)

set\_buses\_names(self, names: list)

set\_transportsType(self, types: list)

get\_transports\_types(self)

get\_money\_cost(self, cost)

get\_power\_cost(self, cost)

get\_time\_cost(self, speed)

class Node

+father : Node

main

Start\_function

+current\_station:station

+goal\_station:station

+person : Person

+time :float

+transport :list +cost :float

+\_\_eq\_\_(self, node: object)

+\_\_lt\_\_(self, other)

class Station

+name:string

+bus\_wait:float

+taxi\_wait:float

\_\_eq\_\_(self, station: object)

set\_name(self, name)

set\_bus\_wait(self, wait)
set\_taxi\_wait(self, wait)

class Person

+money:float

+power:float

+Structure :Structure

\_\_eq\_\_(self, person: object)

set\_money(self, money)

set\_power(self, power)

set\_is\_passenger(self, is\_pass

class Transport(Enum)

+ BUS :Sting

+TAXI :Sting

+ WALK:Sting

sting

# نتائج الحل:

### 1. خوارزمية الـ BFS:

```
Node: A
  Time : 0:00
4 Money: 10000
5 power: 100
6 Transport : ['WALK', 'walk']
10 Node : C
  Time : 0:32:43.636
11
12 Money: 10000
13 power: 70
14 Transport : ['WALK', 'walk']
18 Node : E
  Time : 0:42:32.727
20 Money: 10000.0
21 power: 61.0
22 Transport : ['WALK', 'walk']
26 Node : H
  Time : 1:30:05.454
27
28 Money: 1000.0
29 power: 106.0
30 Transport : ['TAXI', 'taxi']
34 Time Excution: 0:00:00.052
35 Visited node: 73
36 Track node: 4
```

### 2. خوارزمية الـ AStar:

#### اقل كلفة:

```
______
   Node: A
   Time : 0:00
  Money: 10000
  power: 100
   Transport : ['WALK', 'walk']
   ______
  Time: 0:21:49.090
12 Money: 10000
  power: 80
13
   Transport : ['WALK', 'walk']
   ______
   ______
  Node : D
  Time : 0:48:38.181
20 Money: 4800.0
21 power: 106.0
   Transport : ['TAXI', 'taxi']
  ______
  Node : G
  Time : 0:54:05.454
28 Money: 4800.0
  power: 101.0
   Transport : ['WALK', 'walk']
34 Node : F
35 Time : 1:10:27.272
36 Money: 4800.0
  power: 86.0
   Transport : ['WALK', 'walk']
42 Node : H
43 Time : 1:54:05.454
44 Money: 4800.0
  power: 46.0
   Transport : ['WALK', 'walk']
  Time Excution : 0:00:00.015
  Visited node: 40
  Track node: 6
```

```
Node : A
  Time : 0:00
  Money: 10000
  power: 100
  Transport : ['WALK', 'walk']
  ______
  ______
  Time : 0:05
12 Money: 8000
 power: 110
  Transport : ['TAXI', 'taxi']
17
  ______
  Node : D
  Time : 0:10
20 Money: 2800.0
 power: 136.0
  Transport : ['TAXI', 'taxi']
  ______
  ______
  Node : G
Time : 0:15:27.272
28 Money: 2800.0
 power: 131.0
  Transport : ['WALK', 'walk']
  ______
  ______
  Node : F
  Time : 0:21:15.272
36 Money: 2400.0
  power: 123.5
  Transport : ['BUS', 'GF']
  ______
  Node : H
  Time: 0:29:03.272
44 Money: 2000.0
  power: 103.5
  Transport : ['BUS', 'FH']
                Time Excution : 0:00:00.031
 Visited node : 58
 Track node : 6
  ______
```

```
• • •
  Node : A
3 Time: 0:00
4 Money: 10000
5 power: 100
6 Transport : ['WALK', 'walk']
10 Node : C
11 Time: 0:32:43.636
12 Money: 10000
13 power: 70
14 Transport : ['WALK', 'walk']
18 Node : E
19 Time: 0:42:32.727
20 Money: 10000.0
21 power: 61.0
22 Transport : ['WALK', 'walk']
26 Node : H
 Time : 1:30:05.454
27
28 Money: 1000.0
29 power: 106.0
30 Transport : ['TAXI', 'taxi']
32
Time Excution : 0:00:00.001
35 Visited node: 9
36 Track node: 4
```

## تقسيم العمل:

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

## شرح التوابع الرئيسية:

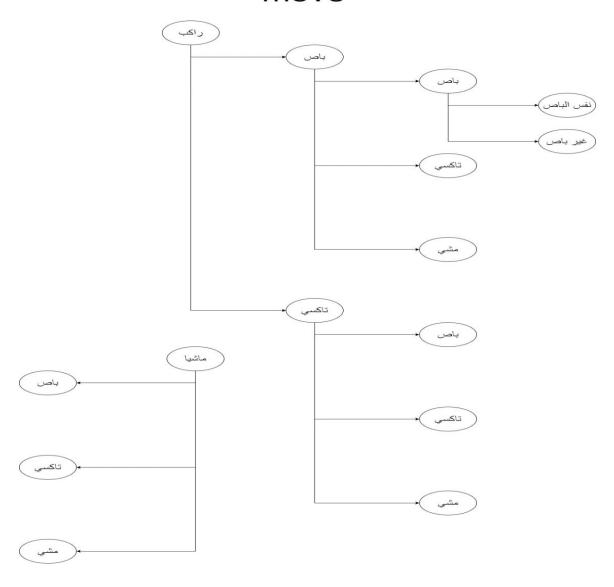
## : Check\_move .1

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 تساوى النقطة الحالية

#### :Move .2

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



- Get\_next\_states .3: تابع يعيد جميع أبناء العقدة الممرة للتابع
- 4. Is\_final\_state: يقوم باخنبار اذا كانت هذه العقدة هي الهدف (المنزل) ام لا
  - get\_track .5 : يقوم بإعادة ليست المسارات التي مر بها الشخص
    - print\_solation .6 يقوم بطباعة حلول الخوارزمية
    - 7. Equal يقوم بتاكد من تساوي عقدتين

# الخوارزمية Astar:

```
def AStar(self):
    now = time.time()
    queue = Queue.PriorityQueue()
    queue.put(self.structure.node)
    self.structure.visited.append(self.structure.node)
    while queue.queue:
        temp_node = queue.queue.pop(0)
    if self.structure.is_final_state(temp_node):
        self.structure.get_track(temp_node)
        break
    for state in self.structure.get_next_states(temp_node):
        if (not self.structure.check_if_in_visited(state)):
            queue.put(state)
        self.structure.visited.append(state)

self.structure.print_soluation(now)
```