

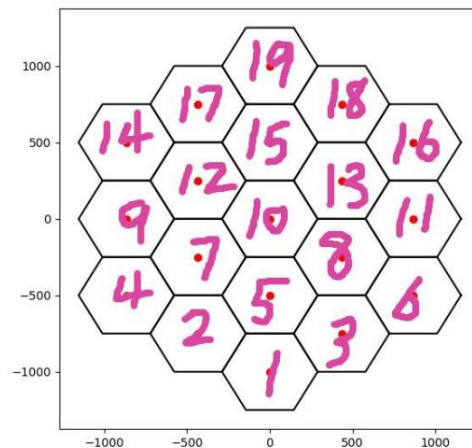
Readme

1. Module 使用方法

把 module.py 跟要 import module 的程式檔放在同一個資料夾下，在程式最上面打
from module import *

2. Functions

map_grid (homework)：畫出地圖的框線 (不含 cell 的點)，呼叫這個 function 後在打 plt.show()，homework 填入 5 或 6，會劃出 7 個 (homework 5) 或者是 19 個 (homework 6) cell 的框框。
MapSetup (BaseStations, homework)：BaseStations 傳入一個空的 dictionary，homework 取決於當前寫的是作業 5 還是 6 (打數字)。這可以把 19 個 cell 都直接設定好，包括裡面的基礎值。cell 編號如下圖：



圖一

getBSpower (homework), getMSpower (), getGain (homework)：用來建立 cell 跟 MS 的，根據 homework 不同回傳 power，你們應該用不到。

其他的 function 都是我在建東西的時候用的，你們應該用不到，有需要再問我。

3. Class Packet & Buffer

Packet 是用來傳資料的單位、Buffer 是 BS/MS 用來暫存的傳不完 Packet 的東西，這兩個你們應該不會直接 call，內容也很簡單，可以看 module 裡面的註解。

4. Class Cell

a. Attributes

x, y：座標，呼叫方法就是 name.x, name.y。

neighbor：一個 dictionary，記錄周邊的 cell，以圖一為例：編號 10 的 neighbor 是 12、15、13、8、5、7。

buffer：一個 Buffer object，原則上你們應該只會 call 到 function，我在 function 裡面應該已經寫好運用了。

id：就當作是 cell 的名字，在 dictionary 呼叫都是用 <dictionary_name>[id] 來呼叫指定 id 的 cell。

homework：使用在 homework 5 或 6。

shan：記錄針對某個 MS uplink 時的 shannon capacity

b. Functions

__init__ (x, y, id, homework)：homework 是 5 或 6。剛創出來的 cell 會沒有 neighbor，使用 setNeighbor 直接加入。shan 初始值為 0，根據要 uplink 的 MS 會更新。

getPower (), getHeight (), setNeighbor (cell), getNeighbor (id), setId (id), setPos (x, y)：用來獲取資訊或設定資訊的。

UPSINR (MS)：傳入要計算 uplink SINR 的 MS，回傳 SINR，這個用在 hw5 中上傳的部分。

UPShannon (MS, divider)：用來計算某一個 MS uplink 給一個 cell 的 Shannon capacity，注意此 function 是從 cell 端呼叫，divider 則需填入平分 bandwidth 的數目，例如一個 cell 可能要接收跟傳送資料，那他能用到 10 MHz 中多少的 bandwidth，需要思考一下，在這邊我只是加上這個東西。

bufferPop (MS)：針對傳入的 MS 檢查 buffer 中有沒有還沒傳完的東西，如果有，會將 buffer 中的資料先消除，再回傳剩下的 Shannon capacity。

bufferUpdate (updateBits, MS)：updateBits 傳入沒有傳完的資料大小，將多的資料放進 buffer 中，會依時間先後順序排列。如果 buffer 滿了，會回傳滿出來的資料大小，即 loss packet。

對於 bufferPop 跟 bufferUpdate 的使用，下方有範例。

5. Class MS

a. Attributes

x, y：座標，呼叫方法就是 name.x, name.y。

cell：負責連線的 cell，可以看成離自己最近的 cell。

id：就當作是 MS 的名字，呼叫都是用 <dictionary_name>[id] 來呼叫指定 id 的 MS。

type：receiver (0) 或是 transmitter (1)，區別任務不同的 MS。

shan：Shannon capacity，一開始為 0，在呼叫過 Shannon 後才會更新。(Shannon 為 MS 的一個 function)

partner：跟自己一組的 MS。

buffer：一個 Buffer object，原則上你們應該只會 call 到 function，我在 function 裡面應該已經寫好運用了。

b. Functions

getPower (), getHeight (), setPos (x, y), setId (id)：獲取或設定資訊。

setPartner (partner, type)：傳入要設為同組的 partner 以及自己是什麼 type

setCell (cell)：傳入自己隸屬於哪個 cell，此 function 會將自己的 position 平移到傳入 cell 的相對座標。如果 setCell 後想要改變座標，使用 setPos。

signal (target)：計算 Receive power，target 可以是 Cell 也可以是 MS (D2D 的 case)。回傳計算結果。

SINR (target, transmitters)：計算 SINR，target 可以是 Cell 或 MS。要注意的是當 target 是 Cell 時，transmitters 不要傳入任何東西；當 target 是 MS 時，transmitters 必須傳入一個記錄著所有 type 為 transmitter 的 MS (就是 type == 1 的)。

Shannon (target, transmitters, divider)：計算針對 target 的 Shannon capacity，回傳計算結果並會自動更新 attributes：shan。target 跟 transmitters 的規則跟前面的 SINR 一樣。divider 這邊有給 default = 1，因為我覺得應該是不用跟別人分 bandwidth，如果使用上你們覺得需要分，那直接給值就可以。

bufferPop, bufferUpdate 跟 Cell 的版本一模一樣，只差在兩個 buffer 大小不一樣。

6. bufferPop & bufferUpdate 使用

```
for mobileID in MS:
    # Transmit data in the buffer first
    ShanCap = BS[10].bufferPop(MS[mobileID])
    # Still have remaining capacity to transmit new data
    if ShanCap > 0:
        if ShanCap > speed:
            ShanCap -= speed
        else:
            old_lossPack = lossPack
            # There's data that is not transmitted
            remainData = speed - ShanCap
            lossPack += BS[10].bufferUpdate(remainData, MS[mobileID])
            lossPack = 0
            loss_occure = 1
ss_prob = (lossPack / (1000 * 1000))
```

圖二

上圖的情境是 MS 要跟 Base station 下載資料，第一行是先將 MS 有的 Shannon capacity 扣掉 cell 的 buffer 中剩下的資料，因為需要 Shannon capacity，在 cell 呼叫 bufferPop 之前需要用先把 MS 針對這個 cell 呼叫 Shannon (cell)，即<MS name>.Shannon(<cell name>, divider)，speed 是每秒資料傳入的速度 (bits/s)。ShanCap 如果小於 speed 代表有資料傳不完需要放到 buffer，用 cell 呼叫 bufferUpdate 來更新，結果回傳 buffer 存不完的資料 (bits/s)，就是 bit loss。

如果是 MS uplink 的情境，把上面 cell 的部分改成 MS，MS 的部分改成 cell 就可以了。