

- 工作環境：  
Ubuntu 16.04 LTS 64-bits on Oracle VM Virtualbox
- 如何 Compile  
按照助教的投影片指令就可以 Compile 了  
>>> make map (產生 ZhuYin-Big5.map)  
>>> make MACHINE\_TYPE=i686-m64 SRIPATH=/home/ta/srilm-1.5.10  
LM=bigram.lm run (compile mydisambig.cpp 且執行)  
>>> make clean (刪除所有執行檔、目的檔)
- 如何 Execute
  - A. 如果已經編譯完成，可以直接輸入  
>> make run  
來產生 result2/1~10.txt
  - B. 如果還沒編譯完成，可以先輸入(參數自行調整)  
>> make map  
>> make MACHINE\_TYPE=i686-m64 SRIPATH=/home/ta/srilm-1.5.10  
LM=bigram.lm  
再執行  
>> make run  
來產生 result2/1~10.txt
  - C. 編譯完成後也可以直接執行  
>> ./mydisambig -text [testdata] -map [mapfile] -lm [lmfile] >  
[outputpath]
- 在這個程式做了甚麼
  - ✧ 在 mapping.py 中，我用 python 簡單寫了一個將 Big5-ZhuYin.map 轉換成 ZhuYin-Big5.map 的程式。主要是用 string 的 strip 函式將一行的頭尾清除，再用 split 將各個字拆開。其中 encoding 注意要用 big5-hkscs，如果用 big5 的話會有些字無法編碼。
  - ✧ 在 mydiambig.cpp 中，我主要是用 std::map 來讀取我的 ZhuYin-Big5.map，然後用 std::vector 把所有注音的可能中文字建立成一個二維 vector，再用 FAQ 上面教的方式來查找 bigram 機率，最後用 viterbi 演算法，透過 dynamic programming 的方式來找出答案。