計算過程

1.計算每間學校在乎條件的**各別分數**

．weight=每間學校在乎條件各別分數陣列

．this.state.n\_distanceBool=使用者是否在乎距離選項，如果在乎，就計算距

離。

this.state.n\_userHome.coordinate=使用者居住城市經緯度

val=學校

val.coordinate =學校的經緯度

算式：透過相減兩點的經緯度，再運用畢氏定理得到兩點之間的長度。

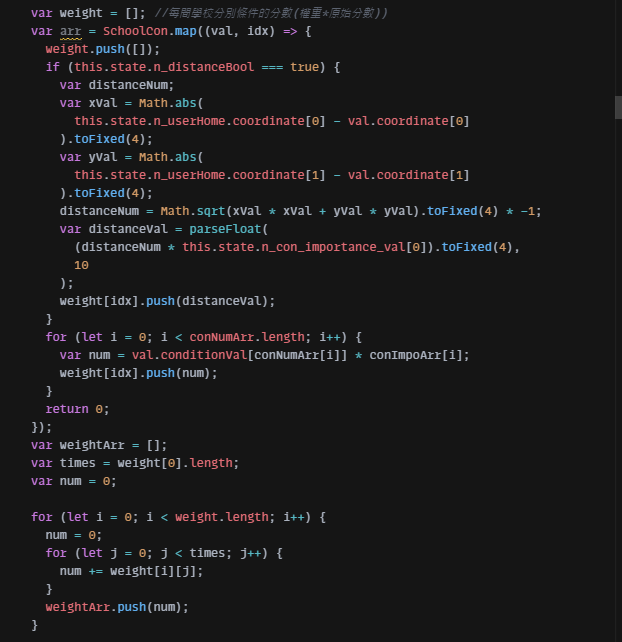
且得到的值要\*-1；因數字越大代表距離越遠，我們最終的結果是看

哪間學校分數最高，因此要\*-1，用扣的方式去計算最後的分數。

．val.condionVal[ ]=學校的各項條件分數

conImproArr=使用者給條件的重要度分數

算式：將以上兩者分數相乘，再放進weight陣列裡

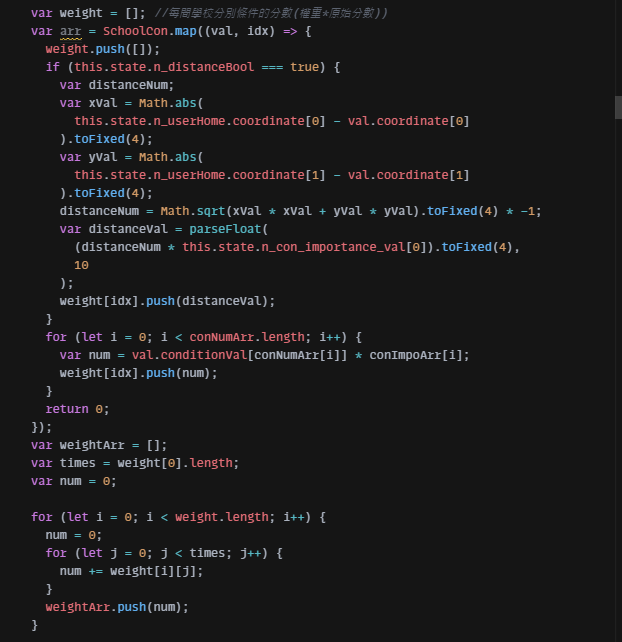


2.計算各間學校最終分數

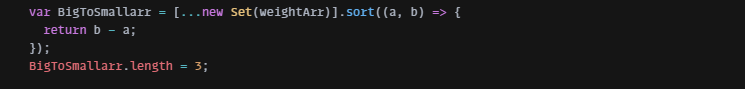
．weightArr=每間學校的分數陣列

算式：將各間學校的各個條件分數進行累加，將條件加完後放進weightArr陣列

裡。



3.排列出分數前三高的學校



4.計算推薦給使用者讀什麼學群什麼系

算式：

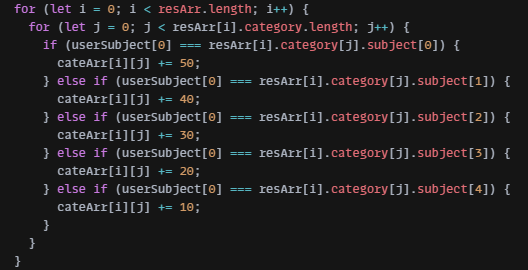
若使用者第一喜歡的科目跟學校的某學群第一在乎的科目一樣 +50 分

若使用者第一喜歡的科目跟學校的某學群第二在乎的科目一樣 +40 分

若使用者第一喜歡的科目跟學校的某學群第三在乎的科目一樣 +30 分

若使用者第一喜歡的科目跟學校的某學群第四在乎的科目一樣 +20 分

若使用者第一喜歡的科目跟學校的某學群第五在乎的科目一樣 +10 分



使用者第二、第三喜歡的科目也是以同樣原理進行運算。

5.得出最高分數的學群為系統最推薦的學群。

