

Task 1: 照著 pseudo code 就好

1-2 的部分我偷了 check_line() 的架構來用，
加上判斷是否為空寶石及更改回傳值為 bool

Task 2: 我先做了一個 elim_gem_special() 用來對指定寶石打標跟遞迴發動能力
剩下的大架構是一樣的：對自己打標，消除自身能力，指定範圍消除
但細節上，若是 Q 發動了 z，Q 要先恢復自身能力這樣 z 才能偵測，
偵測後再重新消除 Q 的能力

Task 3: i 改 9-i 然後 comma

Task 4: 基本邏輯是找到怎麼做能消掉最多的寶石

分三部分：先看特殊，再做特殊，最後普通

第一部分：如果有特殊寶石存在，試著消掉他，

若有複數存在則選擇能消除最多寶石的

第二部分：看看盤面能否做出特殊寶石，優先度：Q > z > +

第三部分：看看怎麼做能消掉最多的寶石(應該會是 3or6)

對於每個回傳值保證他在板內且相鄰

為避免突發狀況若最後傳不出東西會隨機取兩點直到這兩點為合法交換後回傳

第一部分其實跟 splendor 的邏輯差不多，就是 swap 後發動特殊寶石來計算能消除多少，但+不像 Q 跟 z 可以隨便消所以很難做，只能試著列舉所有能消的方法[見 check_cross()]，總而言之最後會有一個列表，裡面會是兩個 Pos 一個 int，代表交換那兩個寶石會消掉幾顆，最後迴圈跑過這張表找最大值。

第二部分意外的簡單，我弄了一個 5x5 deque(其實也可以用 2d-array 但我怕時間不夠)來存部分盤面，這樣做是因為所有的特殊寶石 pattern 必小於 5x5，我不需要每 swap 兩顆就 dfs 整個盤面，感覺就會超時，然後一個 map 存能做出幾顆對應 ABI 的特殊寶石(array 就能做到，這只是炫而已，我怎麼知道會不會有人把資料離散程度弄得很糟糕(X))，最後掃過所有能取的 5x5 版面找特殊。
關於找特殊的方法，基本上就是 swap 兩個寶石，然後 dfs 整個 deque 一輪就好，對，我又偷你的 code 了 owo

第三部分就是，掃過板子一輪執行所有可能的交換後，一樣拿到一個兩個 Pos 一個 int 的表，迴圈找最大值。

下面還有喔~~~

Task Bonus-1: 這裡實作了兩個東西

一、zz 交換會炸了整個版面: 就是字面上的意思, 消除所有版面上的寶石

二、新增一種特殊寶石: 飛機寶石(plane: %)

生成: 當 2x2 範圍內有相同寶石時生成於左上角那格, 不與其他寶石衝突

功能: 字母 % 與相鄰的寶石替換時, 不論是否產生連線皆為有效操作, 替換時會在板上[1~8][1~8]內隨機取一點作為中心, 並消除中心的上下左右及其本身(如同一個小十字), 一樣會遞迴發動其他特殊寶石。

第一項其實比較簡單, 就是 `apply_killsame()` 改一下若 tar 為 z 的行為

第二項因為跟遊戲機制有關要改的就比較多, 要先加上新符號, `apply_special()` 多一種, `eliminate()` 裡面要生成飛機寶石(call `gen_plane()`), `check_eliminate()` 跟 `check_swap()` 要改不然手動模式會出問題。最後最重要的, 要加上 `gen_plane()` 來生成, `apply_plane()` 來實作他的效果。為求方便如果飛機寶石發動他會把 center 印出來讓我們確認有沒有正確發動。

附註: 因為打標的那個誤消 bug, 飛機如果跟 Q 消也會有點 bug

關於 bonus 的東西, 如果 `add_zzswap_and_plane.cpp` 的註解不夠清楚的話可以到我的 github: https://github.com/illumeow/sprout_project1 裡面有一個 bonus 資料夾, 那裡面的 `splendor.cpp` 是我改好能用的

後記:

我應該花了超過 100 小時在作這個 project, 這也是我第一次打這麼多 code, 整個過程有挑戰性, 但 ai 一直出 bug 真的快爆炸, 天天爆肝修。我其實後來有其他想法, 但由於需要打掉重寫我又沒那個時間所以只能繼續下去, 讓整個 code 變得有點疊床架屋, 希望正式上場他可以有好成績。