PA3 大略說說我的想法

電機四 鄧旭辰 b09901017

程序分為幾個部分:

變量定義:

定義了一些輸入和輸出變量，包括圖的類型、頂點數、邊數，以及一個用於存儲邊的向量。

函數定義:

Make_SET、Find_SET、LINK、UNION:這些函數用於實現並查集，一種數據結構，用於高效處理元素分組問題。

use_Kruskal_u_maxST_to_print_remove:使用Kruskal算法處理無向圖，找出最大生成樹，並記錄移除的邊。

d_print_removed_edge_for_directed_graph:處理有向圖，確定移除哪些邊可以保持圖的連通性。

主函數:

讀取輸入文件，根據圖的類型('u' 無向圖，'d' 有向圖)選擇不同的處理方式。

處理完畢後，將結果輸出到指定的輸出文件。

整體上，這段代碼展示了圖的基本處理方法，包括讀取圖信息、應用圖算法，以及根據算法結果進行輸出。它利用了Kruskal算法和深度優先搜索(DFS)來解決圖的連通性和生成樹問題。

而關於這兩個主函數：

use_Kruskal_u_maxST_to_print_remove 函數和

d_print_removed_edge_for_directed_graph 函數是我代碼中的兩個核心部分，它們處理不同類型的圖(無向圖和有向圖)。以下多做一些說明

use_Kruskal_u_maxST_to_print_remove 函數:

- 這個函數運用 Kruskal 算法來處理無向圖。
- 首先對邊進行排序，使得權重較大的邊排在前面。
- 使用並查集方法來檢查和合併不同的連通分量。
- 如果添加一條邊會導致循環，則這條邊會被記錄並移除。
- 最終，函數將生成一個最大生成樹，並記錄被移除的邊。

d_print_removed_edge_for_directed_graph 函數:

- 這個函數專門處理有向圖。
- 它首先使用 上面的u_function 算法對邊進行排序和初始處理。(因為我猜想有向圖的所移除的所有邊，必定都會被包含在無向圖的解答裡，所以先分析無向圖)
- 然後，對於每條候選的移除邊，它使用深度優先搜索（DFS）來檢查移除這條邊後圖是否仍然連通。
- 如果移除後圖不再連通，這條邊會被加回。
- 最終，函數記錄被移除且不影響連通性的邊。