

Project3 Report 0316038 石瑾旋

Implementation :

1. **one_person_joined** : 要插入一個新的元素，就創一個新的 Promenade，然後 call one_group_joined 合併即可，num++。
2. **one_group_joined** : 要合併兩個 Promenade，先把兩個 Promenade 的 BT list 重新串接起來(mergeheap)，以 degree 為 key 做 sorting，再根據這個新的 BT list 開始進行一系列的合併，如果只有兩個高度相同的 BT，就直接合併，如果有三個高度相同的 BT，就把後面兩棵合併(維持 sorting)。
3. **mergeheap** : 要把兩個 Promenade 的 BT list 給重新串接起來，採用 merge sort 的方法，使用 newHead 紀錄合併後的頭，使用 newCurr 來紀錄每次合併後的尾，先尋找誰的開頭比較小，當做新串列的頭，再串起來。
4. **absorb_energy** : 先用 search 找到要 absorb 的 person，如果 absorb 後 energy \leq 0，num--，把他 bubble up 到 root 的位置，然後把 root->child 反序排列，創建一棵新的 Promenade，並且讓他的 head 指向反轉後的 BT list，然後把這個新的 Promenade 跟原本的用 one_group_joined 合併。如果 absorb 後 energy $>$ 0，就把那個 person 跟他的 parent 比，比 parent 小就 bubble up，調整他的位置。
5. **calculate_people_below** : 用 rune (resursive)去紀錄有幾個 person 的 energy \leq energy。
6. **get_weakest_person** : 由於每個 BT 本身都已經是 min-heap 的特性了，因此只要針對每個 BT 的 root 比較其值即可。先讓 min=最大值，然後跟每個 root，如果 root 比 min 小，min=root->energy，然後有一個紀錄最小 person 的指標就指到 root。
7. **size** : 回傳 num 的值。

Sample testing Results :

```
CS Computer Center <help@cs.nctu.edu.tw>
bsd1 [/u/cs/103/0316038] -chshih5747- ls
DSproject_3.pdf Source.cpp      a.out.core      key             right
Promenade.h      Source.exe      g++             leftThread      test1.txt
Promenade.h.gch a.out          is_thread1      public_html     test2.txt
bsd1 [/u/cs/103/0316038] -chshih5747- g++ Source.cpp
bsd1 [/u/cs/103/0316038] -chshih5747- ./a.out test1.txt
The number of people below 5 energy is: 1
The weakest person's id is: 2
The number of people in Elizabeth promenade is: 1
bsd1 [/u/cs/103/0316038] -chshih5747- ./a.out test2.txt
The number of people below 20 energy is: 4
The weakest person's id is: 2
The number of people in Elizabeth promenade is: 7
bsd1 [/u/cs/103/0316038] -chshih5747- █
```