

one_person_joined:

建立一個新的 Promenade 並插入一個 Person，之後和原 Promenade 使用 one_group_joined 結合起來。

one_group_joined:

先將兩個 Promenade 的 head 的 sibling 串起來，再從 head 開始，判斷接下來兩個的 degree 是否相同且第三個 degree 是否與前兩個不同。如果是，則將此兩個 sibling 串起來。

absorb_energy:

修改:開一個陣列記住指標位置(這樣可以做到 $\lg(n)$)，直接修改後和他的 parent 比較，如果比 parent 小，就和他交換。

刪除:若此 person 的 $\text{energy} \leq 0$ ，則他會被交換到最上層，將他的上一個節點的 sibling 直接接到要被移除點的下一個。再將被移除點的 child 以及其 sibling 與原本的 Promenade 合併。

calculate_people_below:

對 sibling 直接遞迴計算，考慮該 Person 的 energy 是否小於等於給定的 energy，如果是，答案加 1 之後遞迴其 child，否則不計算其 child。

get_weakest_person:

針對最上層的 sibling 走訪一次即可知道最小值。

```
[tocknicsu@tocknicsu] - [~/share/oj/lab/104_1/ds/hw3] - [2015-12-31 10:09:57]
[0] <git:(master 4d448c1*) > g++ Source.cpp && ./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
[tocknicsu@tocknicsu] - [~/share/oj/lab/104_1/ds/hw3] - [2015-12-31 10:10:24]
[0] <git:(master 4d448c1*) > g++ Source.cpp && ./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
```