

Judge 3-2

古有皇帝後宮三千，若是要挑一個人來陪都不知道怎麼挑。

有天皇帝想要挑個最瘦的妃子來服伺，卻不知道哪一個才是最瘦的，於是把這個任務交給李公公。聰明的李公公想出了一個方法：妃子分別住在東西十二宮中，先將東六宮跟西六宮的妃子分成東西兩隊，再將兩宮妃子不斷地分成兩小隊，直到每一小隊只有一位妃子。

接著兩兩比對：每一小隊跟相鄰的小隊，按胖瘦排序後合併為一隊，將較瘦的妃子排在前面。接著繼續與相鄰的隊伍比較、排序、合併，直到所有妃子都在同一個隊伍為止。

但是李公公覺得妃子太多，於是找了王公公一起來幫忙。東六宮交給王公公，西六宮就由李公公自己負責。兩邊分別排好後，再將東西兩邊比較完的妃子們從最開頭開始互相比較。那就讓我們一起來幫忙李公公領妃子事寢吧！

輸入：妃子人數

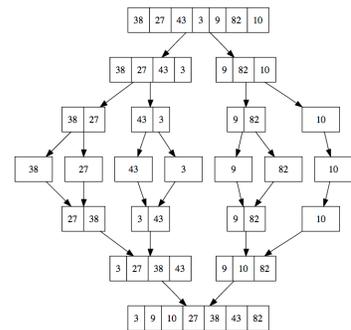
輸出：妃子比較完成排好順序所花費地時間

要求：

- 我們可以把妃子的體重設為整數，用陣列表示，公公代表 thread。
 1. 實作2個公公 (thread)，先把妃子們 (亂數 array) 剖半分給每個公公 (thread) 處理。
 2. 每個 thread (公公) 裡面實作從頭比較並合併的方式。
 3. 合併兩個公公 (thread) 的結果。

提示：合併已排序好的妃子的方法：

1. 申請一個較大的宮殿給新的小隊，使其大小為兩個已經排序妃子小隊之和，該空間用來存放合併後的妃子序列。
2. 設定兩個指標，最初位置分別為兩個已經排序妃子序列的起始位置。
3. 比較兩個指標所指向的妃子體重，選擇相對瘦的妃子放入到合併後的小隊，並移動指標的手指到下一位置。
4. 重複步驟 3 直到某一指標到達序列尾。
5. 將另一序列剩下的所有妃子體重直接複製到合併序列尾。



Reference: wikipedia

- 等到 thread 結束再繼續執行：


```
Thread t = new Thread(...);
t.start();
try {
    t.join();
} catch (InterruptedException ie) {}
// t 結束後才會執行
```
- 計算排序執行時間：


```
long startTime1 = System.currentTimeMillis();
平行選妃(a);
long endTime1 = System.currentTimeMillis();
```

Sample

```
Skye:java sky$ java 選妃 1000
1000 個妃子來選共花了 3 毫秒
辛苦李公公、王公公了！
```