

Problem 1:

每畫錯一條線扣 2 分; 錯三條以上算全錯
只畫出 4-merger 框框, 沒畫內部(扣 6 分)

Problem 2:

每畫錯一條線扣 2 分; 錯三條以上算全錯
shuffle link 方向畫反(扣 4 分)
processors 沒編號(扣 2 分)
exchange link 畫錯(扣 2 分)

Problem 3:

10 分: n processors; $O(1)$ time

5 分: 直接呼叫講義上方法 n processors 找 maximum 兩次, 沒有詳細說明

其他作法:

8 分: double logarithmic depth tree method + cascading

6 分: balanced binary tree method + cascading

4 分: 切 $n^{1/2}$ 區塊, sequential finding maximum: $O(n^{1/2})$ time

將每個區塊的 maximum 拿出來找 global maximum in $O(1)$ time

total time: $O(n^{1/2})$

小錯誤 -1 分

Problem 4:

10 分: 正確說明

6 分: 說明太過於簡略, 且沒有舉例或是例子錯誤

小錯誤 -1 分

Problem 5:

10 分: n processors; $O(\lg n)$ time (Euler-tour technique)

construct an Euler-tour: 2 分

assign weights: 2 分

prefix computation: 2 分

find the depth of each node: 2 分

find the height of tree: 2 分

6 分: 不是 EREW 的作法, 但是結果正確且時間較 sequential 快

(0 分: 平行演算法的時間跟 sequential 一樣)

額外 +2 分: 正確使用 cascading 做到 processors: $O(n / \lg n)$; time: $O(\lg n)$

沒有時間分析 -1

沒有 number of processor -1

小錯誤 -1 分

Problem 6:

(1)

10 分: time: $O(n^{1/2})$, processor: $O(n)$

6 分: 時間比 $O(n)$ 快, 且正確的 parallel algorithm

5 分: 只算出 global sum, 沒有往回修正算出 prefix sum

2 分: 只提概念缺少細節

0 分: 用了無法在 mesh 上執行的 operation

額外 +2 分: 正確使用 cascading 做到 processors: $O(n^{2/3})$; time: $O(n^{1/3})$

(2) 每項各一分:

若回答(1), processor complexity 及 time complexity 照(1)的演算法評分

若無回答(1), processor complexity 及 time complexity 不給分 (因 Problem 3 已提過)

efficiency, cost, speedup 照上述給的 processor complexity 及 time complexity 評分

(3), (4)

0 分: 沒說明原因

Problem 7:

每個技巧算 3 分, 名稱和說明各佔 1.5 分 (說明有小錯誤扣 0.5 分)。

1. balanced binary tree method
2. pointer jumping (doubling)
3. divide and conquer
4. partition
5. pipelining
6. double logarithmic depth tree method
7. accelerate cascading
8. symmetry breaking
9. Brent's Theorem
10. Euler tour technique

11. tree contraction (partial evaluation)
12. sampling
13. multi-level / multi-stage
14. overlapping with computation and broadcasting (duplicate)

以下不算

prune & search, selection

dynamic programming

array, tree, mesh, cube, odd-even merger, bitonic merge sorter, Hypercube, perfect shuffle

broadcasting

rake

以 CW 降低 complexity

k -way search

comparison matrix

semigroup

$n^{1/2}$ -way divide and conquer