

Параллельные сортировки на общей памяти

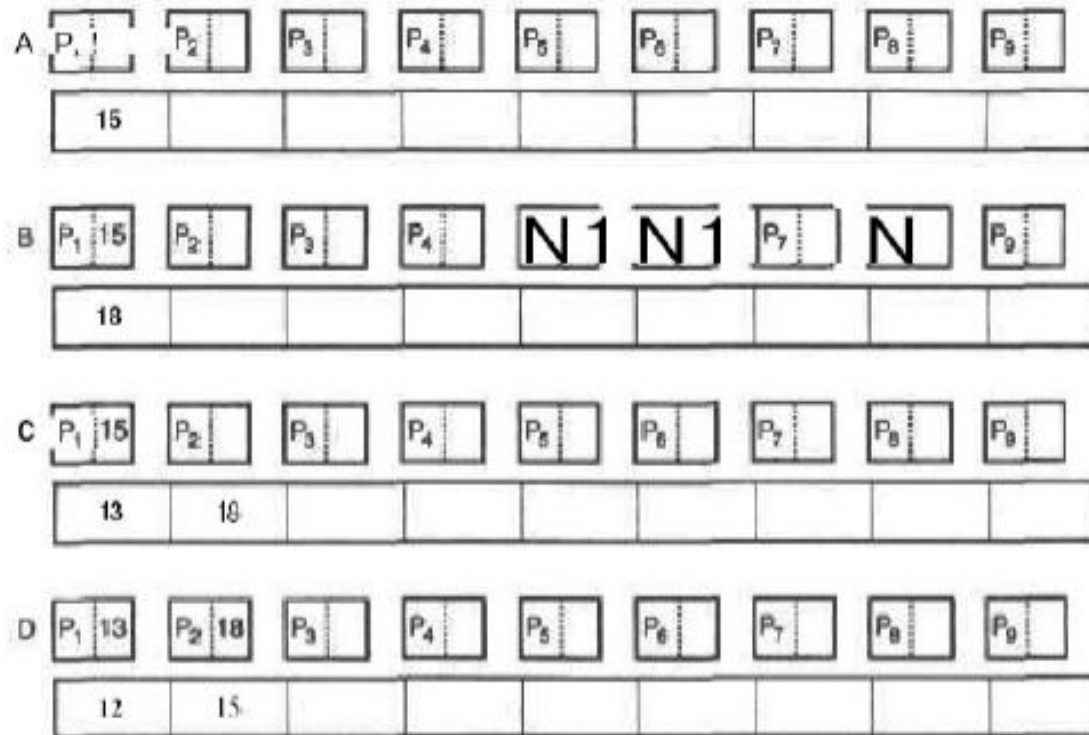
Сортировка на линейных сетях

```
for i=1 to N-1 do
  занести следующее значение в M[1]
  Parallel Start
    P[j] читает M[j] в Current
    for k=1 to j-1 do
      P[k] читает M[k] в New
      if Current>New then
        P[k] пишет Current в M[k+1]
        Current=New
      else
        P[k] пишет New в M[k+1]
      end if
    end for k
  Parallel End
  занести следующее значение в M[1]
  Parallel Start
    P[j] читает M[j] в Current
    for k=1 to j do
      P[k] читает M[k] в New
      if Current>New then
        P[k] пишет Current в M[k+1]
        Current=New
      else
        P[k] пишет New в M[k+1]
      end if
    end for k
  Parallel End
  Parallel Start
    for j=1 to N-1 do
      P[j] пишет Current в M[j]
    end for j
  ParallelEnd
```

Если число процессоров равно числу сортируемых значений, то сортировку можно осуществить, передавая сети в каждом цикле одно значение. Первый процессор читает поданное значение, сравнивает его с текущим и передает его большее значение своему соседу. Остальные процессоры делают то же самое: сохраняют меньшее из двух значений и пересылают большее следующему звену в цепочку.

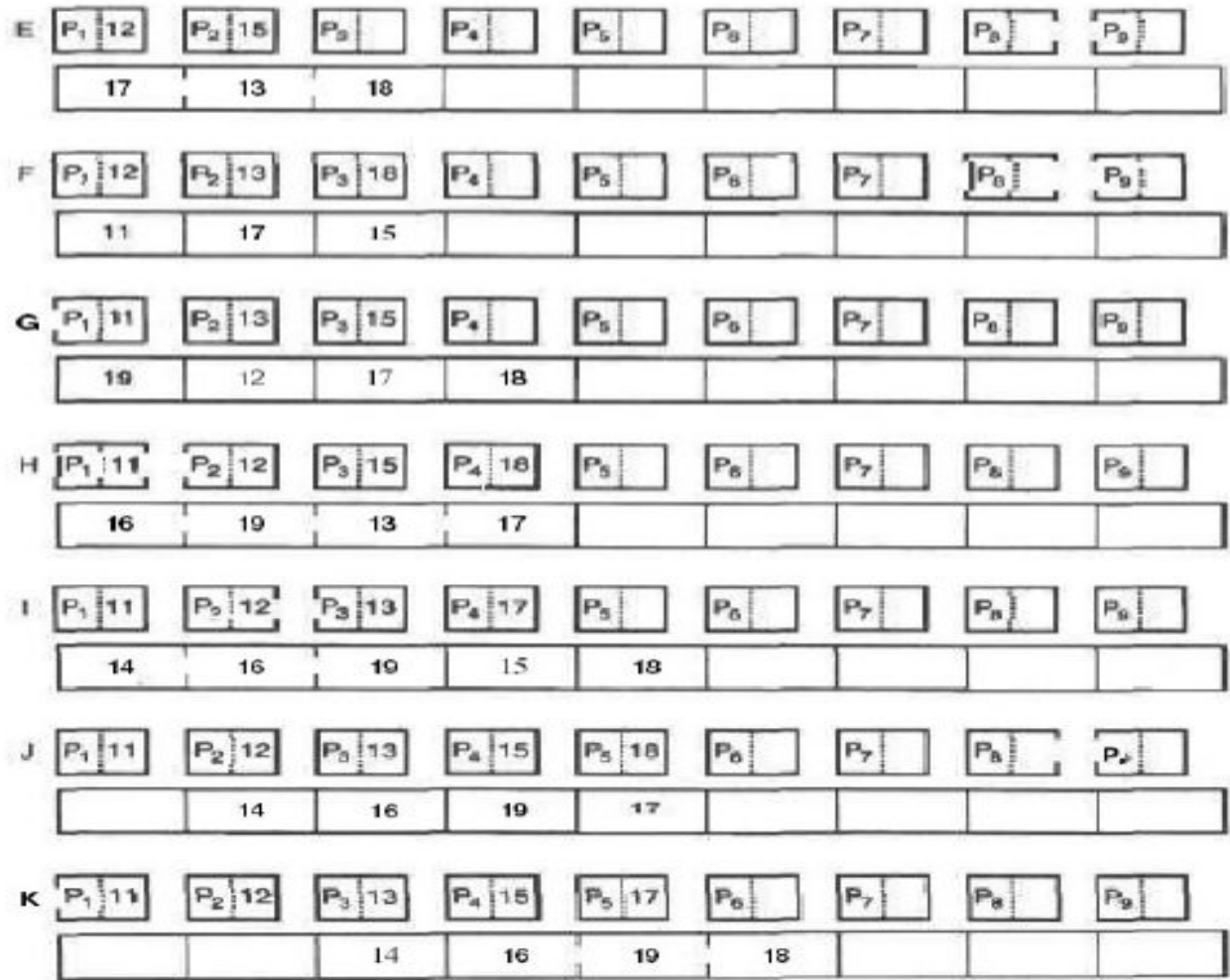
Сортировка на линейных сетях

A.



Сортировка на линейных сетях

В.



Сортировка на линейных сетях

В.



Сортировка на линейных сетях

Характеристики :

- ▶ В общем случае алгоритм выполняет $2 * (N - 1) + 1$ (т.е. $O(N)$), действий.
- ▶ В работе принимают участие N процессоров \Rightarrow стоимость алгоритма равна $O(N)$, что эквивалентно самым медленным нашим сортировкам.

Четно-нечетная сортировка перестановками

```
for j=1 to N/2 do
  Parallel Start
    for k=1 to N/2 do
      P[k] сравнивает M[2k-1] и M[2k]
      if их порядок неправильный then
        переставить их
      end if
    end for k
  Parallel End
  Parallel Start
    for k=1 to N/2-1 do
      P[k] сравнивает M[2k] и M[2k+1]
      if их порядок неправильный then
        переставить их
      end if
    end for k
  Parallel End
end for j
```

В четно нечетной сортировке сравниваются соседние значения и при необходимости переставляются.

Четно-нечетная сортировка перестановками

Пример для списка : 15, 18, 13, 12, 17, 11, 19, 16, 14

	15	18	13	12	17	11	19	16	14
O1	15	18	12	13	11	17	16	19	14
E1	15	12	18	11	13	16	17	14	19
O2	12	15	11	18	13	16	14	17	19
E2	12	11	15	13	18	14	16	17	19
O3	11	12	13	15	14	18	16	17	19
E3	11	12	13	14	15	16	18	17	19
O4	11	12	13	14	15	16	17	18	19
E4	11	12	13	14	15	16	17	18	19

Четно-нечетная сортировка перестановками

Характеристики

- ▶ Все сравнения происходят параллельно, поэтому всякий проход цикла выполняет два сравнения и общее время работы равно $O(N)$.
- ▶ Стоимость алгоритма равна величине $N/2 * O(N)$ — несколько меньше, чем у предыдущего алгоритма, но все равно порядка $O(N^2)$.

Другие алгоритмы

- ▶ Если у нас список без повторений, то мы можем отсортировать его с помощью подсчета. С одним процессором на каждое входное значение местоположение каждого элемента можно определить за $O(N)$ сравнений. Нам требуется N процессоров, поэтому стоимость такого подхода равна $O(N^2)$.
- ▶ Еще существуют способы параллельного слияния двух списков, реализующие оптимальную стоимость $O(N)$. При использовании не более чем $N / \log(N)$ процессоров. Разбив список на $N / \log(N)$ частей, и отсортировав каждую из них эффективным

ВЫВОДЫ

- ▶ Параллельную сортировку можно выполнить различными способами.
- ▶ Если число процессоров равно числу сортируемых значений, то сортировку можно осуществить, передавая сети в каждом цикле одно значение. (Сортировка на линейных сетях)
- ▶ Чтобы уменьшить число процессов вдвое можно воспользоваться следующим способом сортировки, который сравнивает соседние значения и при необходимости переставляет их. (Четно-нечетная сортировка перестановками)