Московский Государственный Университет

имени М.В. Ломоносова

Курс: «Параллельное программирование и суперкомпьютеры»

Расписание сети сортировки

**Выполнил:**

**Сактаганов Нуржан (аспирант)**

**Дата подачи: 23.10.2017**

**Москва 2017**

## Постановка задачи

Разработать последовательную программу вычисления расписания сети сортировки, числа использованных компараторов и числа тактов, необходимых для её срабатывания привыполнении на **n** процессорах. Число тактов сортировки при параллельной обработке не должно превышать числа тактов, затрачиваемых четно-нечетной сортировкой Бетчера.

Параметр командной строки запуска: n. n > 0 – количество элементов в упорядочиваемом массиве, элементы которого расположены на строках с номерами [0…n-1].

Формат команды запуска: **./program n**

Требуется:

1. вывести в файл стандартного вывода расписание и его характеристики в представленном далее формате;
2. обеспечить возможность вычисления сети сортировки для числа элементов 1<=n<=10000;
3. предусмотреть полную проверку правильности сети сортировки для значений числа сортируемых элементов 1<=n<=24;
4. формат файла результата:

n 0 0

cu0 cd0

cu1 cd1

…

cun\_comp-1 cdn\_comp-1

n\_comp

n\_tact

Здесь:

n 0 0 – число сортируемых элементов, ноль, ноль.

cui cdi – номера строк, соединяемых i-м компаратором сравнения перестановки.

n\_comp – число компараторов

n\_tact– число тактов сети сортировки

## Метод решения

На входе имеется массив из n элементов. Алгоритм попарно сравнивает элементы, отстающих друг от друга на степень двойки. Сначала сравниваются наиболее отдаленные. С каждой итерацией это расстояние уменьшается вдвое. На первом шаге это расстояние равно наибольшей степени двойки, меньшей n. На последней итерации сравниваются элементы, отстающие друг от друга на единицу.

## Подсчет тактов

Для подсчета тактов обратимся к физике. Допустим, строим расписание на n элементов. Строим n пронумерованных дорожек (от 0 до n-1). Смотрим на эту схему так, что начала дорожек направлены вниз, а концы вверх (как бы сеть стоит левым боком). Очередной компаратор должен сравнить пару элементов a и b. На компараторы действует гравитация (тянет вниз), а между компараторами действует сила отталкивания (они электрически заряжены). Причем два компаратора могут отталкиваться друг от друга только тогда, когда они находятся на одной и той же линии хотя бы одним концом. Компараторы могут находиться только строго перпендикулярно дорожкам (такая странная физика). Сила электрического отталкивания не позволяет приблизиться компараторам меньше чем на единицу расстояния. А гравитация — отдалиться больше чем на единицу расстояния. Постепенно туда сверху кладем компараторы, и смотрим на высоту полученной конструкции. Она и определит количество тактов.

## Метод проверки

Проверка каждого расписания для 1 <= n <= 24 производится принципом нулей и единиц. Т.е. берутся все возможные комбинации нулей и единиц заданной длины n – а их всего два в степени n. Над каждой такой комбинацией производятся операции сравнения и обмена согласно проверяемому расписанию. После выполнения расписания над текущей комбинацией, проверяется, что она отсортирована. Расписание считается корректной, если все комбинации нулей и единиц заданной длины в итоге окажутся отсортированы в результате выполнения расписания.

Эту проверку можно запустить передав третий необязательный параметр. Например:

**program 0 k**

Здесь k указывает до какой длины (включительно) проверяется алгоритм построения расписания.