# PRL – projekt 1 Odd-Even Merge Sort

Autor: Jakub Smejkal (xsmejk28)

## Analýza

Analýza tohoto algoritmu je složitá vzhledem k větvení kaskády sítí a narůstajícímu počtu procesorů, potřebných k vyřešení větších vstupních polí.

Podmínkou pro funkčnost algoritmu bez přidávání či ubírání hodnot je, aby byl počet prvků na vstupu mocninou čísla dvě. Důvodem je, že každý procesor má dva vstupy => n = 2m

Písmeno "m" v této rovnici značí počet fází potřebných pro seřazení celého vstupního pole o "n" prvcích.

Fází se rozumí podčást sítě procesorů, která postupně zpracovává vstup. Kde v 1. fázy je potřeba 2m-1 procesorů, ve 2. fázy je potřeba 2m-2 sítí 2x2 po třech procesorech, atp.

Na grafu níže je vidět závislost počtu prvků na počtu procesorů pro demonstraci.

Časová složitost je dána => t(n) = O(m2) = O(log2n)

Celková cena algoritmu se vypočítá jako => c(n) = t(n) \* p(n)

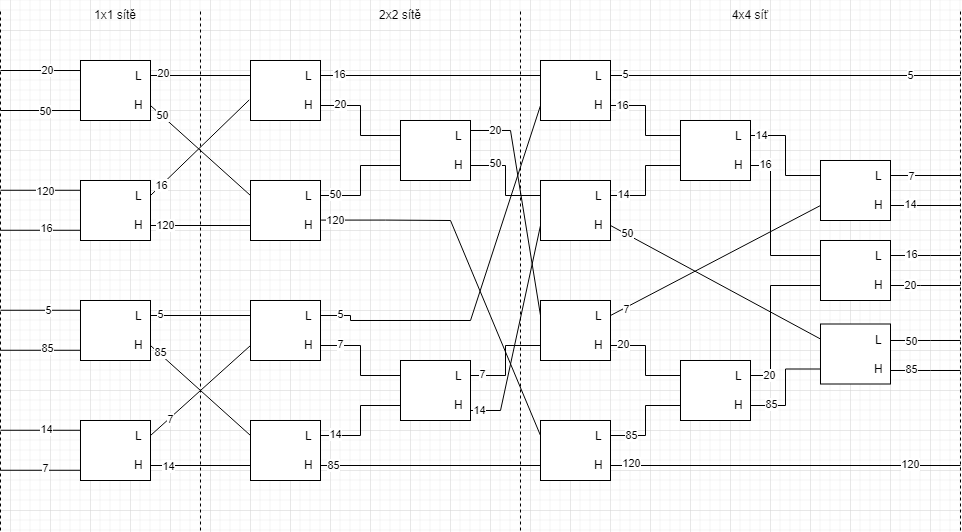
Na grafu níže je zobrazena celková cena algoritmu při rostoucí velikosti vstupního pole.

## Implementace

Ukázková implementace byla vytvořena pro 8 náhodně vygenerovaných hodnot. Pro těchto 8 hodnot je nutné mít několik sítí zapojených za sebou.

Jsou použity 3 úrovně, kde v první úrovni jsou čtyři 1x1 sítě, v druhé úrovni jsou dvě 2x2 sítě a ve třetí a poslední úrovni je jedna 4x4 síť. Toto uspořádání vyžaduje 4 procesy v první, 6 procesů v druhé a 9 procesů ve třetí úrovni. Celkový počet procesů potřebný pro tuto implementaci je tedy 19.

Na obrázku níže je možné vidět ukázku propojení jednotlivých procesů i se vzorovými vstupními hodnotami, které jsou touto sítí seřazeny.



Samotný kód je prováděn v několika jednoduchých krocích:

* Inicializace MPI
* Pokud se jedná o proces s rankem 0 (Master proces) tak načte hodnoty ze souboru a rozešle všem procesům v první části sítě (1x1) každému po 2 hodnotách
* Každý proces po obdržení obou vstupních hodnot provádí funkci oddEvenMerge(), kde je podle ranku procesu rozhodnuto, kterému procesu bude odeslána menší a kterému větší hodnota
* Procesy odesílající hodnoty na výstup odesílají své hodnoty Master procesu s příslušným Tagem, který určuje pořadí hodnoty ve výstupním poli
* Po obdržení všech hodnot, je seřazené pole vypsáno na výstup
* Konec programu