

## Параллельный Merge Sort с использованием библиотеки OPENMP

n - количество данных, m - максимальный размер чанка, P - число потоков

n = 3000000, m = 10000, P = 1

Qsort time: 0.637012s

Merge sort time: 0.759214s

n = 3000000, m = 10000, P = 2

Qsort time: 0.631601s

Merge sort time: 0.374244s

n = 3000000, m = 10000, P = 4

Qsort time: 0.633185s

Merge sort time: 0.263806s

n = 3000000, m = 10000, P = 8

Qsort time: 0.621235s

Merge sort time: 0.308580s

n = 3000000, m = 10000, P = 16

Qsort time: 0.654020s

Merge sort time: 0.322809s

```
In [9]: import matplotlib.pyplot as plt
X = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
Y1 = [0.759214, 0.374244, 0.393784, 0.263806, 0.293163, 0.305734, 0.293180,
      0.327422, 0.326234, 0.320402, 0.329517, 0.322809]
Y2 = [0.637012, 0.631601, 0.651485, 0.633185, 0.636493, 0.632300, 0.658972,
      0.642487, 0.636607, 0.633302, 0.633664, 0.654020]

Y = [Y2[i]/Y1[i] for i in range(len(Y1))]
Z = [Y[i]/X[i] for i in range(len(Y1))]

plt.plot(X, Y1, label = "Parallel Merge Sort")
plt.plot(X, Y2, label = "qsort")
plt.title("T(P)")
plt.grid(True)
plt.legend()
plt.show()

plt.plot(X, Y, label = "Acceleration")
plt.title("S(P)")
plt.grid(True)
plt.legend()
plt.show()

plt.plot(X, Z, label = "Scale")
plt.title("E(P)")
plt.grid(True)
plt.legend()
plt.show()
```



