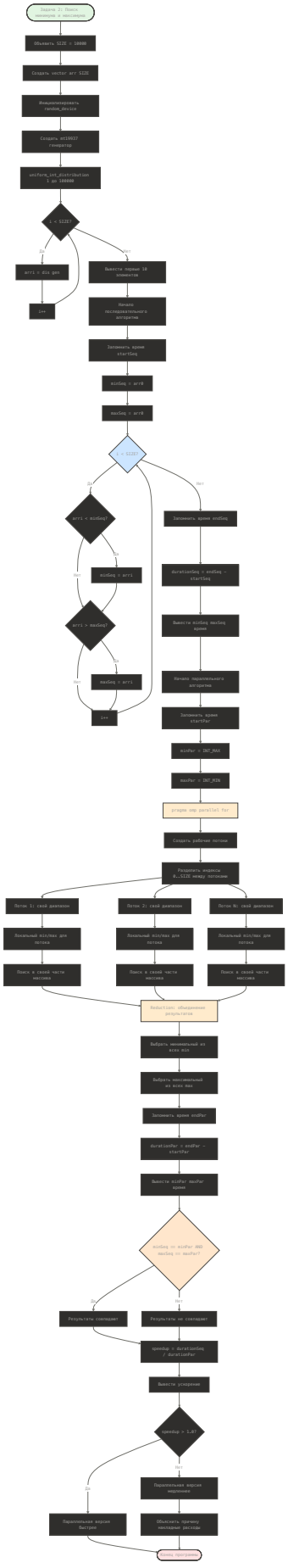
Задача 1: Поиск минимума и максимума
Объявить SIZE = 100000
Создать vector arr SIZE
Инициализировать random_device
Создать mt19937 генератор
uniform_int_distribution 1 до 100000
i < SIZE?
Да
arr[i] = dis(gen)
i++
Нет
Вывести первые 10 элементов
Начало последовательного алгоритма
Запомнить время startSeq
minSeq = arr[0]
maxSeq = arr[0]
i < SIZE?
arr[i] < minSeq?
minSeq = arr[i]
arr[i] > maxSeq?
maxSeq = arr[i]
i++
Запомнить время endSeq
durationSeq = endSeq - startSeq
Вывести minSeq maxSeq время
Начало параллельного алгоритма
Запомнить время startPar
minPar = INT_MAX
maxPar = INT_MIN
pragma omp parallel for
Создать рабочие потоки
Распределить индексы 0...SIZE между потоками
Поток 1: свой диапазон
Поток 2: свой диапазон
Поток N: свой диапазон
Локальный minElem для потока
Локальный maxElem для потока
Локальный minElem для потока
Поиск в своей части массива
Поиск в своей части массива
Поиск в своей части массива
Reduction: объединение результатов
Выбрать минимальный из всех min
Выбрать максимальный из всех max
Запомнить время endPar
durationPar = endPar - startPar
Вывести minPar maxPar время
minSeq == minPar AND maxSeq == maxPar?
Да
Результаты совпадают
Нет
Результаты не совпадают
speedup = durationSeq / durationPar
Вывести ускорение
speedup > 1.0?
Да
Параллельная версия быстрее
Нет
Параллельная версия медленнее
Объяснить причину накладных расходов
Конец программы