

1. Заметим, что случаи при $n \leq 3$ надо рассмотреть отдельно для дальнейшей оптимизации.
2. t прибавляется всегда при i от 0 до $n/2$. Можно вынести за цикл, тем самым сократив кол-во операций в цикле.
3. Оставшийся цикл можно разбить теперь на два цикла: от 2 до $n/2$ и от $n/2$ до n . Избавились от конструкции `ifelse`.
4. В первом цикле вызывается `bar(i)`, причем $i > 1$, значит `bar()` всегда вернет i . Этот цикл можно убрать, используя формулу суммы членов арифметической прогрессии.
5. Во втором цикле учтем, что вызывается `bar(i+t)`. t может быть отрицательным. Начнем вызывать `bar()`, начиная с того элемента, сумма которого с t будет больше 1. Оставшуюся сумму (если она есть) вычислим по формуле суммы членов арифметической прогрессии.
6. В итоге полностью избавился от цикла, заменив его математическими операциями. Количество операций, выполняемых при этом процессором уменьшилось очень сильно.
7. **Время исполнения программы (на исходных данных) уменьшилось с примерно 20с до примерно 0.005с (скриншот замера времени прилагается). Причем, компилируя исходный код с -O3, результат был порядка 3с**
8. Изначально было желание сделать `bar()` и `foo()` `static inline` функциями, но компилятор отказывался делать это (возможно, с его точки зрения это только усложнит ассемблерный код).