

РАЗРАБОТКА РЕКУРСИВНЫХ АЛГОРИТМОВ

Цель работы: Разработка программ, реализующих различные рекурсивные алгоритмы, и оценка их временной и пространственной сложности.

Словесное описание алгоритма: рекурсивный алгоритм для перевода чисел из десятичной системы в двоичную либо восьмеричную.

Текст программы:

```
if (decimal_number <= 0 || numSystem > 10) {  
    return 0;  
}  
else {  
    return (decimal_number % numSystem + 10 *  
            convertToOtherNumSystem( decimal_number: decimal_number / numSystem, numSystem));  
}
```

Формулы верхней оценки временной и ёмкостной сложности:

Трудоемкость рекурсивных алгоритмов зависит как от количества операций, выполняемых при одном вызове функции, так и от количества таких вызовов. Так как рекурсию можно заменить циклом, то порядок временной сложности соответствующего алгоритма можно считать равным числу повторений цикла. В моем методе выполняется 4 операции + блок ветвления.