Зробити оцінку складності функцій за кодом:

int func1(int i) {

if (i == 0) {

return 1;

}

return i\*func1(i-1);

}

Відповідь:

Часова складність: O(n) - функція є рекурсивною і викликає сама себе зі зменшенням значення 'i', поки не досягне 0. Це призводить до 'n' рекурсивних викликів, де 'n' - вхідне значення 'i'.

Просторова складність: O(n) – функцію є рекурсивною і викликає сама себе із зменшенням значення ‘i’, поки не досягне 0. Кожного разу, коли функція func1 викликається рекурсивно, створюється новий кадр стеку, який поміщається у стек викликів. В результаті, буде створено n кадрів стеку.

void func(std::vector<std::vector<int>>& v, int i) {

if (i == 0) {

return;

}

func(v, i/2);

std::vector<int> r;

for (int j = 0; j < i; j++) {

r.emplace\_back(func1(i)/func1(j)/func1(i-j));

}

}

Відповідь:

Часова складність: Функція func1 має часову складність O(n), де n - вхідне значення i. Функція func викликає func1 i разів у циклі, що призводить до часової складності O(n^2).

Просторова складність: O(n) - для кожного рекурсивного виклику створюється вектор r розміру i. Отже, загальна складність простору за рахунок рекурсивних викликів у func дорівнює O(i).