

1. Опишите процесс компиляции программы.
2. Определения: $f(n) \in O(g(n))$, $f(n) \in \Omega(g(n))$, $f(n) \in o(g(n))$
3. Упорядочите, что во что вложено в определении O -асимптотике:

$$n; \log n; n^2; n^{\frac{3}{2}}; n^n; 10; 2^n; n!.$$

4. Найдите ассимптотику по времени исполнения, а также укажите количество выделенной памяти в зависимости от длины входной строки для следующего кода:

```
#include <iostream>
#include <string.h>

const char number = 0xFFFF;

int function(const char* str){
    printf("%s\n", str);
    if(strlen(str) > 0){
        function(str+1);
    }
    return 0;
}

int main(){
    char a[number];
    // вводится строка длины n
    scanf("%s", a);
    function(a);
    return 0;
}
```

5. Какие основные элементы классов?
6. Укажите все известные вам структуры данных.

Таблица 1

Структура	insert(index)	pop(index)	find(index)
List			
Array			

7. Что такое стек, очередь? В чем отличие?

8. Заполните таблицу 1 асимптотическим временем выполнения операций в терминах $O(n)$, где n — это количество элементов в структуре данных.

9. Опишите алгоритм, для нахождения минимального пути в таблице 2 из правой верхней клетки в нижнюю левую клетку, если можно идти только влево или вниз.

Таблица 2

0	7	2	5	9	2	7
4	6	6	6	4	3	8
3	6	4	4	3	8	7
3	2	7	2	2	7	9
4	5	8	2	6	5	6

10. Определение графа и дерева.

11. Продемонстрируйте обход графа при помощи поиска в ширину для графа на рис. 1. Начальные вершины выбираете на свое усмотрение.

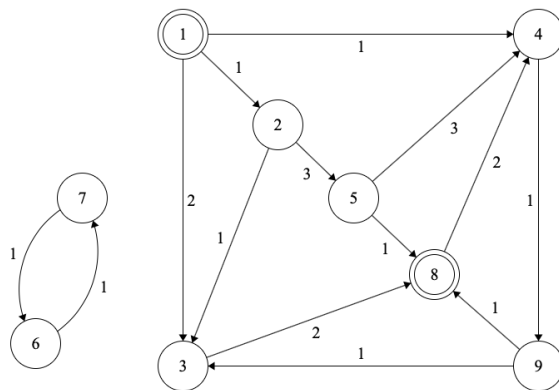


Рис. 1

12. Продемонстрируйте обход графа при помощи поиска в глубину для графа на рис. 1. Начальные вершины выбираете на свое усмотрение.

13. Продемонстрируйте обход графа при помощи поиска в глубину для графа на рис. 1. Начальные вершины выбираете на свое усмотрение.

14. Продемонстрируйте работу алгоритма Дейкстры для графа на рис. 1. Начальная вершина 1. Укажите расстояние между отмеченными вершинами.