

## Результаты

СД 5+20 мин

03 июля 2025 в 13:02 – 03 июля 2025 в 13:14

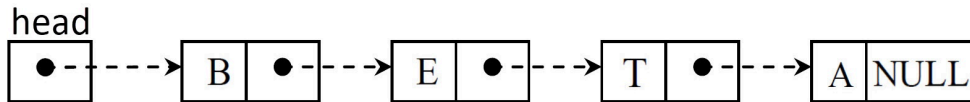
ДЕ 1

Тема: Линейные списки

1.

12516261

Какая структура данных представлена на рисунке:



- Односвязный линейный список без указателя
  - Односвязный линейный список с фиктивным звеном
  - Односвязный линейный список с указателем
  - Циклический линейный список без указателя
- (один вариант)

1 из 1

2.

12516262

Определите назначение заданной функции:

```
struct list
{
    float data;
    list *next;
};

float function(list *head)
{
    int i;
    list *curr;
    float sum = 0;
    curr = head;

    for (i = 0; curr != NULL; i++)
    {
        sum += curr->data;
        curr = curr->next;
    }
    return sum / i;
}
```

- Вычисление среднего арифметического всех элементов односвязного линейного списка, начиная с head
  - В тексте программы есть ошибка, функция не будет работать
  - Вычисление среднего арифметического всех элементов нелинейного списка, начиная с head
  - Вычисление среднего арифметического всех элементов односвязного линейного списка
- (один вариант)

1 из 1

3.

12516263

Определите назначение заданной функции:

```
int function(list *p, int x)
{
```

```

list *r;
while (p->next != NULL && p->next->data != x)
    p = p->next;
if (p->next)
{
    r = p->next;
    p->next = r->next;
    delete r;
    return 1;
}
else return 0;
}

```

- Удаление по значению в ациклическом списке с фиктивным звеном
  - Удаление по значению в ациклическом списке без фиктивного звена
  - Удаление по значению в циклическом списке с фиктивным звеном
  - Удаление по значению в циклическом списке без фиктивного звена
- (один вариант)

1 из 1

4.

12516264

Структуры данных – список. Определите назначение заданной функции:

```

list *function (list *p, float x)
{
    if (p == NULL) return NULL;
    if (p->data == x) return p;
    return function(p->next, x);
}

```

- Поиск элемента по значению
  - Поиск пустого элемента
  - Включения элемента x в список
  - Поиск непустого элемента
- (один вариант)

1 из 1

Итог по теме: 4 из 4

Тема: Линейные списки (профильный уровень)

1.

12516265

Выберите правильные утверждения:

- По односвязному списку можно двигаться в двух направлениях
- Чтобы работать со списком, как с единым объектом, нужно ввести в употребление статическую переменную-указатель, значение которой – это адрес первого (или заглавного) звена списка
- В циклическом односвязном списке поле ссылки последнего элемента имеет значение NULL
- Если первое звено имеет ссылку на последнее, а последнее – на первое, то список называют кольцевым (циклическим)

(возможно нескольких вариантов)

2 из 2

2.

12516267

Структуры данных – список. Преимуществами связанного отображения перед несвязным являются:

- Требуется меньшее количество памяти (для хранения указателей, связывающих элементы)
  - Легко изменить список: для включения или исключения элемента достаточно изменить связи между элементами
  - Возможно образовать сложную и даже нелинейную структуру
  - Последовательный доступ к элементам
- (возможно нескольких вариантов)

2 из 2

Итог по теме: 4 из 4

Итого по дидактической единице: 8 из 8

ДЕ 2

Тема: Стек и очередь

1.

12516269

Линейный список, в котором доступен только один элемент, называется:

- Массивом
- Стеком
- Очередью
- Деком

(один вариант)

1 из 1

2.

12516270

В каком случае нужно использовать стек или очередь?

- Если при удалении и включении элемента требуется смещение элементов
- Если все действия над элементами выполняются в соответствии с их ключами
- Если возникает необходимость уменьшения затрат по времени
- Если при обработке множества данных нельзя обработать текущие данные

(один вариант)

1 из 1

3.

12516271

Если положить в очередь символы “D”, “C”, “B”, “A” в указанном порядке и затем по одному извлечь их из очереди, то порядок извлечения будет:

- ABDC
- DCAB
- ABCD
- DCBA

(один вариант)

1 из 1

4.

12516273

Какое из приведенных ниже утверждений относится к структуре данных **очередь**?

Механизм, реализующий правило: "первым вошел – последним вышел" (first in – last out)

Это список, у которого доступны два элемента (две позиции): начало и конец очереди

Структура, в которой каждый элемент содержит указатели на другие элементы списка

Это список, у которого доступен один элемент (одна позиция) (один вариант)

1 из 1

Итог по теме: 4 из 4

Тема: Стек и очередь (профильный уровень)

1.

12516278

Что из нижеперечисленного относится к понятию очередь с приоритетом?

Такой очереди наилучшим образом отвечает двусвязный ациклический список

Элемент с наивысшим приоритетом помещается в конец очереди

При включении элемента в очередь сначала определяется место элемента в соответствии с его приоритетом

Такой очереди наилучшим образом отвечает двусвязный циклический список (возможно нескольких вариантов)

2 из 2

2.

12516280

За что отвечают подпрограммы function1, function2 представленные ниже:

```
struct stack
{
    int top;
    type data[n];
}
```

```
int function1(char *x, stack *st)
{
    if (empty(*st))
    {
        *x = st->el[st->top];
        st->top--;
        return 1;
    }
    return 0;
}
```

```
int function2(char x, stack *st)
{
    if (st->top != n - 1)
    {
        st->top++;
        st->el[st->top] = x;
    }
}
```

```
    return 1;
}
else return 0;
}
```

- Подпрограмма извлечения элемента из статического стека
- Подпрограмма добавления элемента в статический стек
- Подпрограмма проверки статического стека на пустоту
- Подпрограмма извлечения элемента из динамического стека  
(возможно нескольких вариантов)

2 из 2

Итог по теме: 4 из 4

Итого по дидактической единице: 8 из 8

ДЕ 3

Тема: Бинарные деревья

1.

12516281

Что представляет собой прямой обход бинарного дерева?

- Корень, левое поддерево в прямом порядке, правое поддерево в прямом порядке
- Левое поддерево в прямом порядке, корень, правое поддерево в прямом порядке
- Левое поддерево в прямом порядке, правое поддерево в прямом порядке, корень
- Корень, правое поддерево в прямом порядке, левое поддерево в прямом порядке

(один вариант)

1 из 1

2.

12516284

Что делает данная подпрограмма:

```
void function(stack *s, btree*d)
{
    while (d != NULL )
    {
        printf_s("%d ", d->elem);

        if (d->left != NULL && d->right != NULL )
        {
            push_back(s, d->right);
            d = d->left;
        }

        else if (d->left == NULL && d->right == NULL)
        {
            if (s->top != -1)
            {
                d = s->data[s->top];
                pop_back(s);
            }
        }
    }
}
```

```

    }
    else d = NULL ;
}

else if (d->left != NULL )
    d = d->left;

else d = d->right;
}
}

```

- Распечатывает все элементы дерева в порядке прямого обхода
- Распечатывает все элементы дерева в порядке обратного обхода
- Распечатывает все элементы дерева, помимо концевых, в порядке прямого обхода
- Распечатывает все элементы дерева, помимо концевых, обходя его в глубину

(один вариант)

1 из 1

3.

12516286

На сколько может отличаться число вершин в левом и правом поддеревьях каждой вершины бинарного идеально сбалансированного дерева?

- Не менее, чем на 1
- На 2
- Не более, чем на 1
- На 1

(один вариант)

1 из 1

4.

12516287

Что делает данная подпрограмма:

```

void function(stack *s, btree*d)
{
int c = 0;
while (d != NULL)
{
if (d->left != NULL && d->right != NULL)
{
push_back(s, d->right);
d = d->left;
}

else if (d->left == NULL && d->right == NULL)
{
c++;
if (s->top != -1)

```

```
{
d = s->data[s->top];
pop_back(s);
}
else d = NULL;
}

else if (d->left != NULL)
d = d->left;

else d = d->right;
}
printf_s("%d ", c);
}
```

- Распечатывает количество листов в дереве в порядке прямого обхода
- Распечатывает количество листов в дереве в порядке обратного обхода
- Распечатывает количество вершин в дереве в порядке обратного обхода
- Распечатывает количество вершин в дереве в порядке прямого обхода  
(один вариант)

1 из 1

Итог по теме: 4 из 4

Тема: Бинарные деревья (профильный уровень)

1.

12516289

Какими из перечисленных свойств обладает бинарное дерево поиска:

- Для любого поддеревя значение правого потомка больше значения корня
- Для любого поддеревя значение левого потомка больше значения корня
- Для любого поддеревя значение левого потомка меньше значения корня
- Для любого поддеревя значение правого потомка меньше значения корня  
(возможно нескольких вариантов)

2 из 2

2.

12516291

Какие из высказываний верные?

- Степень вершины определяется количеством дуг, выходящих из данной вершины
- Число поддеревьев данного узла называется степенью этого узла
- Число поддеревьев, образованных на уровне данной вершины, называется степенью узла
- Степень вершины определяется количеством дуг, входящих в эту вершину  
(возможно нескольких вариантов)

2 из 2

Итог по теме: 4 из 4

Итого по дидактической единице: 8 из 8

ДЕ 4

Тема: Управление таблицами

1.

12516293

В чем заключается метод деления для получения хеш-функции?

- Ключу ставится в соответствие целая часть деления целого значения ключа на  $N-1$ , где  $N$  – размер таблицы
- Ключу ставится в соответствие остаток от деления целого значения ключа на  $N$ , где  $N$  – размер таблицы
- Ключу ставится в соответствие целая часть произведения  $N$  (размер таблицы) на  $M$ , где  $M$  – дробная часть произведения целого значения ключа на константу из интервала  $(0, 1)$
- Ключу ставится в соответствие дробная часть произведения  $N$  (размер таблицы) на  $M$ , где  $M$  – дробная часть произведения целого значения ключа на константу из интервала  $(0, 1)$

(один вариант)

1 из 1

2.

12516294

В чём заключается метод умножения для получения хеш-функции?

- Ключу ставится в соответствие дробная часть произведения  $N$  (размер таблицы) на  $M$ , где  $M$  – дробная часть произведения целого значения ключа на константу из интервала  $(0, 1)$
- Ключу ставится в соответствие целая часть произведения  $N$  (размер таблицы) на  $M$ , где  $M$  – дробная часть произведения целого значения ключа на константу из интервала  $(0, 1)$
- Ключу ставится в соответствие целая часть деления целого значения ключа на  $N-1$ , где  $N$  – размер таблицы
- Ключу ставится в соответствие остаток от деления целого значения ключа на  $N$  – размер таблицы

(один вариант)

1 из 1

3.

12516297

В чём заключается главное преимущество хеш-таблиц над остальными структурами данных?

- Быстрый поиск

- Простота реализации
- Требуется меньше затраты памяти
- Быстрая вставка и удаление

(один вариант)

1 из 1

4.

12516300

Определите назначение заданной функции:

```
bool func(table *Table, int x)
{
    int i = 0;
    for ( ; Table->el[i].m < x && i < Table->n - 1; i++);
    if (Table->el[i].m == x)
    {
        while (i < Table->n - 1)
        {
            Table->el[i] = Table->el[i + 1];
            i++;
        }
        Table->n--;
        return true;
    }
    return false;
}
```

- Удаление элемента x из упорядоченной таблицы
- Включение элемента x в таблицу с открытой адресацией
- Включение элемента x в таблицу с использованием бинарного поиска
- Исключение элемента x из неупорядоченной таблицы

(один вариант)

1 из 1

Итог по теме: 4 из 4

Тема: Управление таблицами (профильный уровень)

1.

12516301

При включении элемента в хеш-таблицу используются функции:

- Функция расстановки
- Функция перестановки
- Функция условной выборки

Функция вторичной расстановки  
(возможно нескольких вариантов)

2 из 2

2.

12516303

Как называют хеш-таблицу, в которой можно разрешить коллизию?

- Распределяемая таблица

- Таблица цепочек
- Дополнительная таблица
- Таблица с перемешиванием  
(возможно нескольких вариантов)

2 из 2

Итог по теме: 4 из 4

Итого по дидактической единице: 8 из 8

ДЕ 5

Тема: Алгоритмы упорядочивания

1.

12516306

Какой из методов сортировки является улучшенным вариантом сортировки прямого включения?

- Быстрая сортировка
- Сортировка Шелла
- Сортировка пузырьком
- 

Блочная сортировка

(один вариант)

1 из 1

2.

12516309

Какой из методов сортировки является улучшенным вариантом или разновидностью сортировки пузырьком?

- Быстрая сортировка
- Сортировка включением
- Сортировка выбором
- Пирамидальная сортировка

(один вариант)

1 из 1

3.

12516311

Что делает данная подпрограмма:

```
int n, a[n]; //n - количество элементов
```

```
void sort(int *s_arr, int first, int last)
```

```
{  
    if (first < last)  
    {  
        int left = first, right = last, middle = s_arr[(left + right) / 2];  
        do  
        {  
            while (s_arr[left] < middle) left++;  
            while (s_arr[right] > middle) right--;  
            if (left <= right)  
            {  
                int tmp = s_arr[left];  
                s_arr[left] = s_arr[right];  
                s_arr[right] = tmp;
```

```

        left++; right--;
    }
} while (left <= right);
qs(s_arr, first, right);
qs(s_arr, left, last);
}
}

```

- Сортирует методом пирамидальной сортировки в порядке убывания
- Сортирует методом быстрой сортировки в порядке убывания
- Сортирует методом пирамидальной сортировки в порядке возрастания
- Сортирует методом быстрой сортировки в порядке возрастания  
(один вариант)

1 из 1

4.

12516312

Ниже написана подпрограмма сортировки пузырьком, укажите состояния массива [1, 4, 5, 3, 2, 8, 4, 8, 3] после 10-й итерации:

```

void function(int *a, int n)
{
    int stage = 1;
    for (int i = n - 1; i >= 0; i--)
        for (int j = 0; j < i; j++)
            if (a[j] > a[j + 1])
            {
                int tmp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = tmp;
            }
}

```

- [1, 3, 2, 4, 5, 4, 8, 3, 8]
- [1, 4, 3, 2, 5, 8, 4, 8, 3]
- [1, 4, 3, 2, 5, 4, 8, 8, 3]
- [1, 3, 4, 2, 5, 4, 8, 3, 8]

(один вариант)

1 из 1

Итог по теме: 4 из 4

Тема: Алгоритмы упорядочивания (профильный уровень)

1.

12516313

Укажите компараторы (функции сравнения) для восходящей сортировки:

- `function comparator(a, b) { return a || b; }`
- `function comparator(a, b) { return a && b; }`
- `function comparator(a, b) { return a > b; }`
-

```
function comparator(a, b) { return a >= b; }
```

(возможно нескольких вариантов)

2 из 2

2.

12516314

Какие методы сортировки основаны на сортировке обменом?

- Быстрая сортировка
- Сортировка пузырьком
- Сортировка Шелла
- Пирамидальная сортировка

(возможно нескольких вариантов)

2 из 2

Итог по теме: 4 из 4

Итого по дидактической единице: 8 из 8

Итог по тесту: 40 из 40