**Сортировки**

1. **Сортировка пузырьком:**

for(i=0; i<size-1; i++)

{

for(j=(size-1); j>i; j--)

{

if(Mass[j]<Mass[j-1])

{

temp=Mass[j];

Mass[j]=Mass[j-1];

Mass[j-1]=temp;

}

}

}

1. **Сортировка выбором:**

for(i=0; i<size-1; i++)

{

min=i;

for(j=i+1; j<size; j++)

{

if(Mass[j]<Mass[min])

{ min=j;

}

temp=Mass[i];

Mass[i]=Mass[min];

Mass[min]=temp;

}

}

1. **Сортировка вставками:**

for(i=1; i<size; i++)

{

for(j=i; Mass[j-1]>Mass[j]; j--)

{

temp=Mass[j-1];

Mass[j-1]=Mass[j];

Mass[j]=temp;

}

}

1. **Шейкер Сортировка**

while(left<=right)

{

for(i=size-1; i>=left; i--)

{if(Mass[i-1]>Mass[i])

{

int temp=Mass[i];

Mass[i]=Mass[i-1];

Mass[i-1]=temp;

left++;}

}

for(i=1; i<=right; i++)

{

if(Mass[i-1]>Mass[i])

{

int temp=Mass[i];

Mass[i]=Mass[i-1];

Mass[i-1]=temp;

right--;}

}

}

1. **Сортировка Шелла**

for (int step=size/2; step>0; step/=2)

{

for (int i=size; i>=step; i--)

{

if (Mass[i]<Mass[i-step])

{

int temp=Mass[i];

Mass[i]=Mass[i-step];

Mass[i-step]=temp;

} } }

1. **Быстрая сортировка**
2. **Сортировка подсчётом**
3. **Сортировка Слиянием**
4. **Пирамидальная сортировка**

**Алгоритмы поиска**

1. **Функция последовательного прямого поиска**

#include <stdbool.h>

bool p=false;

for (int i=0; i<size; i++)

{

if(Mass[i]==value)

{

printf("%d ", i);

p=true;

}

}

if(p==false)

{return -1; }

1. **Функция бинарного поиска**

int i, left=0, right=N, middle=(left+right)/2;

for(i=0; i<N/2; i++)

{

if(Mass[middle]==key){printf("Index=%d", middle);; break;}

else if(key>Mass[middle]){left=middle; middle=(left+right)/2;}

else {right=middle; middle=(left+right)/2;}

1. **Функция поиска методом транспозиции**

for (int i = 0; i<size; i++)

{

if (Mass[i]==value)

{

temp=Mass[i];

Mass[i]=Mass[i-1];

Mass[i-1]=temp;

printf("%d ", i);

p=true;

}

}

if(p==false)

{return -1;}

1. **Функция поиска с перемещением в начало**

for (int i = 0; i<size; i++)

{

if (Mass[i] == value)

{

temp = Mass[i];

Mass[i] = Mass[0];

Mass[0] = temp;

printf("%d ", i);

p=true;

}

if(p==false)

{return -1;}

}

1. **Функция индексно-последовательного поиска**