

Двоичный поиск

«...хотя первый двоичный поиск был опубликован в 1946 году, первый двоичный поиск не содержащий ошибок был опубликован только в 1962...»

Дональд Кнут

Угадай число

- Загадано число от 1 до 100, используя подсказки больше-меньше угадайте число.
- За какое число попыток можно гарантированно угадать число?

Угадай число

За какое число попыток можно гарантированно угадать число из интервала от 1 до 100?

1 попытка	из 100 чисел осталось 50
2 попытка	из 50 – 25
3 попытка	из 25 – 13
4 попытка	из 13 – 7
5 попытка	из 7 – 4
6 попытка	из 4 – 2
7 попытка	из 2 - 1

Угадай число

За какое число попыток можно гарантированно угадать число из интервала?

интервал	попыток
1 - 100	7
1 - 1000	10
1 - 10000	$\log_2 10000 < 14$

Двоичный (бинарный) поиск

метод деления пополам - самый быстрый поиск по упорядоченному набору данных

1. Задать левую границу поиска
2. Задать правую границу поиска
3. Найти середину интервала
4. Сравнить середину с искомым значением
5. Перейти в левую или правую часть в зависимости от результата сравнения
6. Повторить с первого пункта

ДВОИЧНЫЙ ПОИСК

Заданы номера выигрышных лотерейных билетов. Определить является ваш билет

выигрышным или нет?

0	1	2	3	4	5	6	7	8	9	10	11	12	l	r	m
2	6	9	20	21	29	35	42	67	96	123	345	454	0	12	6
2	6	9	20	21	29	35						454	6	12	9
2	6	9	20	21	29	35			96			454	6	9	7
2	6	9	20	21	29	35	42		96			454	7	9	8
2	6	9	20	21	29	35	42	67	96			454	7	8	

ДВОИЧНЫЙ ПОИСК

```
l=0; r=n;
while (l<r)
{
    m=(l+r)/2;
    if (a[m]<k) l=m;
    else r=m;
}
if (a[l]==k) // найден
else // не найден
```

ДВОИЧНЫЙ ПОИСК

(с поддержкой инварианта)

```
l=0; r=n;           // полуинтервал [0; n)  
инвариант  
while (l+1<r)  
{  
    m=(l+r)/2;  
    if (a[m]<=k) l=m; // инвариант [ ; )  
    else r=m;  
}  
if (a[l]==k) // найден // инвариант [ ; )  
else // не найден
```

ДВОИЧНЫЙ ПОИСК

(с повторяющимися элементами)

1 1 1 3 3 3 5 5 5 7 7 8 8 8

```
l=0; r=n;  
while (l+1<r)  
{  
    m=(l+r)/2;  
    if (a[m]<=k) l=m;  
    else r=m;  
}  
if (a[l]==k)           // найден  
else                 // не найден
```

находит самое правое вхождение

ДВОИЧНЫЙ ПОИСК

(с повторяющимися элементами)

1 1 1 3 3 3 5 5 5 7 7 8 8 8

```
l=-1; r=n-1;  
while (l+1<r)  
{  
    m=(l+r)/2;  
    if (a[m]<k) l=m;  
    else r=m;  
}  
if (a[r]==k)           // найден  
else                   // не найден
```

находит самое левое вхождение