

Академия Государственной противопожарной службы МЧС России

Кафедра информационных технологий

Дисциплина «Технологии программирования»

Лекция

Тема «Алгоритмы поиска. Основные понятия и классификация»

к.т.н., заместитель начальника кафедры ИТ УНК АСИТ
майор вн. сл. Мокшанцев А.В.

3 корпус, кабинет 611

Москва 2020 г.

Содержание дисциплины на 1-й семестр 2014/2015 учебного года

- ▶ 1. **Алгоритмы поиска**
- ▶ 2. Алгоритмы генерации комбинаторных объектов
- ▶ 3. Анализ алгоритмов и структуры данных
- ▶ 4. Комплексное индивидуальное задание (защита - 6 и 18.12.2014 г.)
- ▶ 5. Экзамен (19 января 2015 г.)

Учебные вопросы

- ▶ 1. Основные понятия и определения алгоритмов поиска.
- ▶ 2. Классификация алгоритмов поиска
- ▶ 3. Последовательный поиск
- ▶ 4. Двоичный (бинарный поиск)
- ▶ 5. Интерполяционный поиск

Основные понятия и определения алгоритмов поиска

Алгоритм поиска – это алгоритм, который в заданном файле ищет запись с заданным ключом (то что необходимо найти).

распространенных операций при обработке данных.

Разнообразие условий, при которых производится

поиск, определяет существование большого

Основные понятия и определения алгоритмов поиска

алгоритмами
сортировки, мы

работаем с
данными,
разделенными на

Цель поиска —

элементы
базы данных

элементов с
ключами,

которые

соответствуют
заданному ключу
поиска.

Назначением
поиска является
получение

доступа к
информации с
целью ее

Примеры поиска



поиска
кументов

формации о пожарах и ЧС

- Поиск информации об авиарейсах
- Поиск информации о материально-техническом обеспечении



Классификация алгоритмов поиска

внешние (по способу размещения)

Статические и динамические

Основные на сравнении

ключей или на цифровых

Использующие сами ключи

или их образы

1. Алгоритмы, Классификация алгоритмов поиска сравнения

Ключей:

- методы поиска
в
последовательно
организованных
структурах;

2. Алгоритмы поиска в древовидных структурах, использующие цифровые свойства ключей.

3. Алгоритмы использующие образы ключей.

Основные понятия и определения алгоритмов поиска

Поиск, при котором файл располагается в основной памяти, называется **внутренним**

поиском
Если же часть или весь файл располагается во внешней памяти, поиск называется **внешним**.



Основные понятия и определения алгоритмов поиска

Поиск называется *статическим*, если содержимое файла не меняется.

Динамический ПОИСК предполагает, что в файл вставляются элементы или удаляются из него.

12.1	0.13	-1,5	0	21.9	-3.7	5.0	121.7
1-й элемент массива	2-й элемент массива	3-й элемент массива	4-й элемент массива	5-й элемент массива	6-й элемент массива	7-й элемент массива	8-й элемент массива

Удаление 2-го элемента



Вставка 6-го элемента



12.1	0.13	-1,5	0	21.9	-3.7	5.0	121.7
1-й элемент массива	2-й элемент массива	3-й элемент массива	4-й элемент массива	5-й элемент массива	6-й элемент массива	7-й элемент массива	8-й элемент массива

Поиск в последовательно организованных структурах

Последовательный поиск

путем
последовательного
сравнения
очередного
рассматриваемого
значения с
искомым до тех

Время выполнения

- ▶ Если отрезок имеет длину n , то найти решение с точностью до ε можно за время n / ε

Сложность

- ▶ Сложность линейного поиска $O(n)$

Основные понятия и определения алгоритмов поиска

Сложность

- мера использования алгоритмом ресурсов времени или пространства.

Время выполнения алгоритма определяется количеством тривиальных шагов, необходимых для решения проблемы и зависит от размера входных данных (далее n).

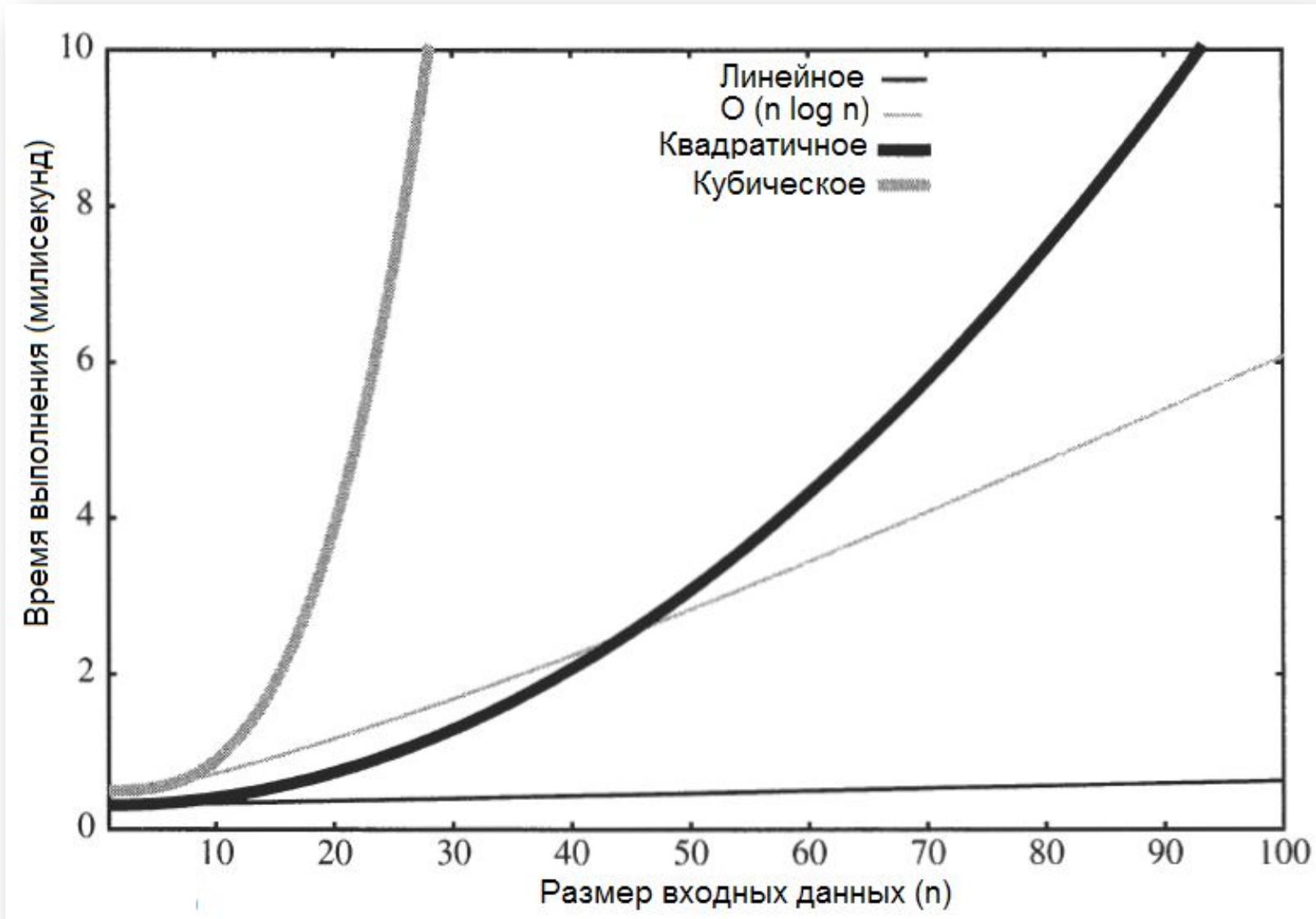
Пространство

- объём памяти или места на носителе данных, используемом алгоритмом.

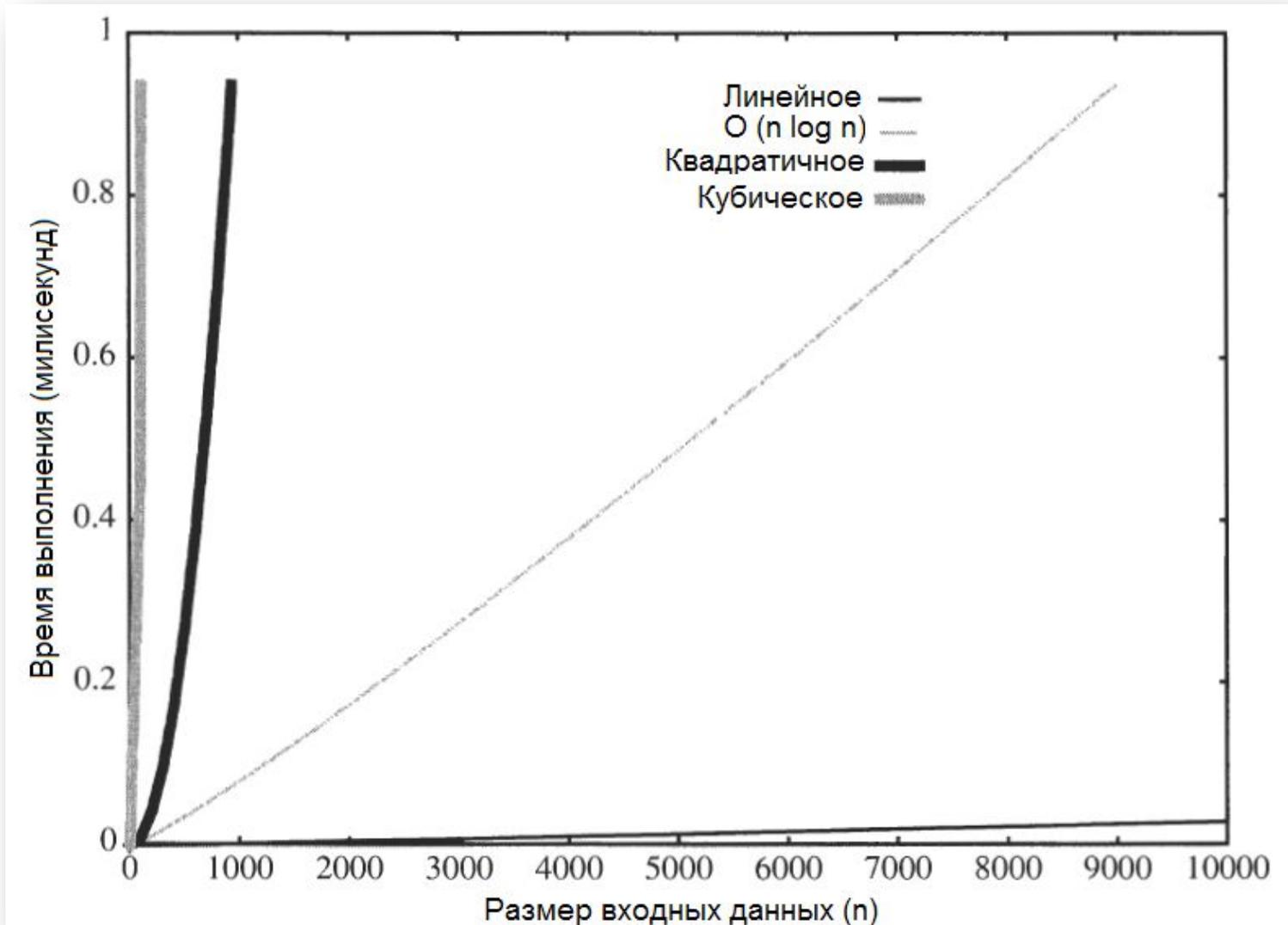
Классы оценок сложности

- множества вычислительных проблем, для решения которых известны алгоритмы, схожие по сложности
 - $O(1)$ – постоянное время
 - $O(\log(n))$ – логарифмическое время
 - $O(n)$ – линейное время
 - $O(n \log(n))$ – "n-log-n" время
 - $O(n^2)$ – квадратичное время
 - $O(n^3)$ – кубическое время
 - $O(2^n)$ – экспоненциальное время

Время выполнения алгоритма для небольших n



Время выполнения алгоритма для больших n



Последовательный поиск на языке C++

Фу
нкц
ия
по
с
л
е
д
о
в
а
т
е
л
ь
н
о
г
о

Код C++

```
1 // Поиск ключа в массиве
2 // функция возвращает индекс искомого элемента
3 // В случае отсутствия искомого элемента возвращает -1.
4 int find(const int* arr, int size, int key ) {
5     for(int i=0; i < size; ++i)
6         if(arr[i] == key)
7             return i;
8
9     return -1;
10 }
```

Последовательный поиск на языке C++

Пример программы последовательного поиска.

Код C++

```
1  #include <iostream>
2
3  // Листинг, представленный выше
4
5  int main()
6  {
7      const int size = 100;
8      int arr[size];
9
10     for(int i = 0; i < size; ++i)
11         arr[i] = i + 1;
12
13     int key = 50;
14
15     int index = find(arr, size, key);
16
17     std::cout << "Element " << key << " found at index " << index << std::endl;
18
19     system("pause");
20
21     return 0;
22 }
```

Последовательный поиск

Преимущества:

- Не требует сортировки значений множества
 - Не требует дополнительного анализа функции.
 - Не требует дополнительной памяти.
- => Следовательно, может работать в потоковом режиме при непосредственном получении данных из любого источника.

Недостатки:

- Малоэффективен по сравнению с другими алгоритмами поиска.
- => Следовательно, используется, если множество содержит *небольшое количество элементов*

Поиск в последовательно организованных структурах

Двоичный (бинарный) поиск

- поиск заданного элемента на упорядоченном множестве, осуществляемый путем неоднократного деления этого множества на две части таким образом, что искомый элемент попадает в одну из этих частей. Поиск заканчивается при совпадении искомого элемента с элементом- границей между частями множества или при отсутствии искомого элемента.

ленном в
виде
массива,
является
двоичны

Время выполнения

- ▶ Когда функция имеет вещественный аргумент - $(\log a)$, где $a=1/\varepsilon$.

Сложность

- ▶ Сложность бинарного поиска $O(\log n)$



Дж. Мокли

Описание метода бинарного поиска

1. Упорядоченное по возрастанию множество элементов, необходимо найти элемент со значением, равным 9

1	5	9	13	21	22	31	46	50
---	---	---	----	----	----	----	----	----

2. Выбор середины вектора – элемента-границы

1	5	9	13	21	22	31	46	50
---	---	---	----	----	----	----	----	----

Описание метода бинарного поиска

3. 1. Сравнение элемента-границы с искомым элементом: $9 < 21$, отбрасываем правую часть

1	5	9	13	21	22	31	46	50
---	---	---	----	----	----	----	----	----

4. 2. В левой части повторяем алгоритм до тех пор, пока элемент-граница не равен 9

1	5	9	13	21	22	31	46	50
---	---	---	----	----	----	----	----	----

Бинарный (двоичный) поиск на языке C++

Пример программы
бинарного поиска.

```
1 #include "stdafx.h"
2 #include <iostream>
3 using namespace std;
4 const int N=10;
5 //двоичный поиск
6 int BinarySearch(int A[], int key)
7 {
8     int left=0, right=N, mid;
9     while (left<=right)
10    {
11        mid=left+(right-left)/2;
12        if (key<A[mid]) right=mid-1;
13        else if (key>A[mid]) left=mid+1;
14        else return mid;
15    }
16    return -1;
17 }
18 //главная функция
19 void main()
20 {
21     setlocale(LC_ALL,"rus");
22     int i, key;
23     int A[N];
24     cout<<"Искомый элемент > "; cin>>key; //ввод ключа
25     cout<<"Исходный массив: ";
26     for (i=0; i<N; i++) //заполнение и вывод массива
27     { A[i]=N*i; cout<<A[i]<<" "; }
28     if (BinarySearch(A, key)==-1) cout<<"\nэлемент не найден";
29     else cout<<"\nномер элемента: "<<BinarySearch(A, key)+1;
30     system("pause>>void");
31 }
```

Двоичный (бинарный) поиск

Преимущества:

- Относительная быстрота выполнения поиска (по линейным)

Недостатки:

- Бинарный поиск может применяться только на упорядоченном множестве

Поиск в последовательно организованных структурах

Интерполяционный поиск

В основе интерполяционного поиска лежит операция интерполирование. *Интерполирование* – нахождение промежуточных значений величины по имеющемуся дискретному набору известных значений. Интерполяционный поиск работает только с упорядоченными массивами; он похож на бинарный, в том смысле, что на каждом шаге вычисляется некоторая область поиска, которая, по мере выполнения алгоритма, сужается. Но в отличие от двоичного, интерполяционный поиск не делит последовательность на две равные части, а вычисляет приблизительное расположение ключа (искомого элемента), ориентируясь на расстояние между искомым и текущим значением элемента.

Сложность

- ▶ Сложность интерполяционного поиска
 $O(\log(\log(N)))$

Поиск в последовательно организованных структурах

Интерполяционный поиск

Работа этого метода также аналогична предыдущему, с той только разницей, что на каждом шаге рассматривается элемент, находящийся не в середине текущей части массива, а в позиции, которая находится от начальной на расстоянии, определяемом по формуле:

$$p = \frac{(r - l)(k - a_l)}{a_r - a_l}$$

Здесь r и l — соответственно правая и левая граница области поиска, k — искомый элемент, a_l и a_r - соответственно левый и правый элемент области поиска.

Поиск в последовательно организованных структурах

Интерполяционный поиск

Рассмотрим пример. Пусть дан массив, упорядоченный по возрастанию.

2	4	5	8	9	16	22	28	36
---	---	---	---	---	----	----	----	----

Пусть требуется найти число $k=22$. Это число больше крайнего левого и меньше крайнего правого элементов. Следовательно, существует возможность найти его в массиве.

В данном примере $l=1$, $r=9$ (областью поиска является весь массив от первого до 9-го элемента включительно, $a_l=2$, $a_r=36$)

Подставим данные в формулу для вычисления номера элемента.

$$P = (9 - 1) * (22 - 2) / (36 - 2);$$

$$P = 8 * 20 / 34;$$

$$P = 4,7; \text{ (для определенности округлим в большую сторону, } P=5\text{).}$$

Поиск в последовательно организованных структурах

Интерполяционный поиск

Рассмотрим элемент с номером $I+P=6$. Этот элемент равен 16 и он меньше искомого числа. Следовательно, поиск необходимо продолжить в правой части массива (теперь I будет равно 6, $a_I=16$).

2	4	5	8	9	16	22	28	36
---	---	---	---	---	----	----	----	----

Вычислим шаг по формуле

$$P=(9 - 6)*(22 - 16) / (36 - 16);$$

$$P=3*6/20;$$

$$P=0,9 \text{ (округлим } P \text{ в большую сторону, } P=1).$$

Рассмотрим элемент номер 7. Он равен искомому числу, поиск окончен.

Интерполяционный поиск на языке C++

Пример программы
интерполяционного
поиска.

```
1 #include "stdafx.h"
2 #include <iostream>
3 using namespace std;
4 const int N=17;
5 //интерполяционный поиск
6 int InterpolSearch(int A[], int key)
7 {
8     int mid, left=0, right=N-1;
9     while (A[left]<=key && A[right]>=key)
10    {
11        mid=left+((key-A[left])*(right-left))/(A[right]-A[left]);
12        if (A[mid]<key) left=mid+1;
13        else if (A[mid]>key) right=mid-1;
14        else return mid;
15    }
16    if (A[left]==key) return left;
17    else return -1;
18 }
19 //главная функция
20 void main()
21 {
22     setlocale(LC_ALL, "Rus");
23     int i, key;
24     int A[N]={2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59};
25     cout<<"Искомый элемент > "; cin>>key; //ввод ключа
26     cout<<"Исходный массив: ";
27     for (i=0; i<N; i++) cout<<A[i]<<" "; //вывод массива
28     if (InterpolSearch(A, key)==-1) cout<<"\nЭлемент не найден";
29     else cout<<"\nНомер элемента: " <<InterpolSearch(A, key)+1;
30     system("pause>>void");
31 }
```

Интерполяционный поиск

Преимущества:

- Если значения данных распределены достаточно равномерно, то он обеспечит наилучшую производительность.

Недостатки:

- Как правило, алгоритм используется лишь на очень больших массивах (внешний поиск)



Контрольные вопросы



- ▶ 1. Что называется алгоритмом поиска?
- ▶ 2. Что называется внутренним и внешним поиском?
- ▶ 3. Какие виды поиска существуют?
- ▶ 4. Сложность алгоритма поиска - это?
- ▶ 5. Эффективен ли последовательный поиск при большом количестве данных?

Литература



C/C++ и MS Visual C++ 2010 для начинающих / Б. Пахомов



Технологии и методы программирования: учеб. пособие для студ. учреждений высш. проф. образования / Н.В. Анашкина, Н.Н. Петухова, В.Ю. Смольянинов. — М.: Издательский центр «Академия», 2012. — 384 с.



Visual C++ 2010
Полный курс / Айвор Хартон

Интернет

<http://cppstudio.com>

<http://kvodo.ru>

<http://www.cyberforum.ru>

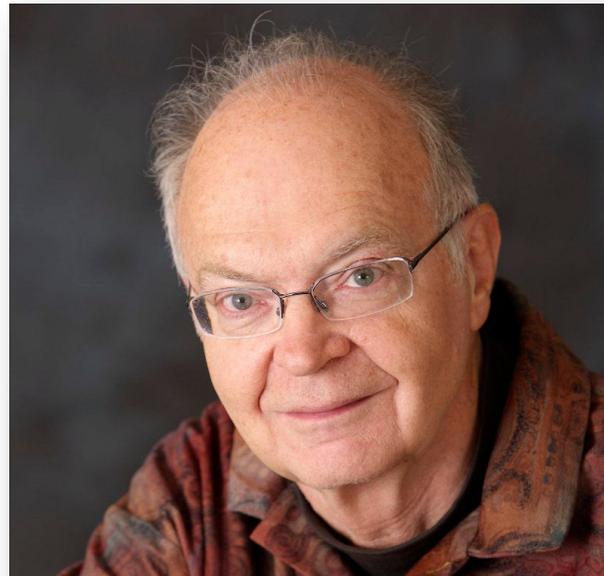


Задание на самостоятельную подготовку

- ▶ 1. Изучить на основе примеров виды поиска.
- ▶ 2. Подготовить ответы на контрольные вопросы.



«Часто говорят, что человек ничего не понимает, пока не объяснит это кому-то другому. Я бы перефразировал это так: человек глубоко не понимает предмет до тех пор, пока не научит этому компьютер, т.е. выразит что-либо в виде алгоритма... Попытка формализовать нечто в виде набора алгоритмов приводит к более глубокому пониманию сути вещей.»



Дональд Кнут