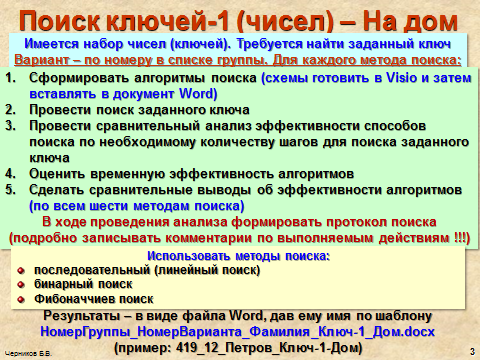
Ф.И.О: **Хоанг Хай –** Группа БИ 4110 – Номер варианта: 34

**Задание по поиску 4110-2-Дом**

****

|  |  |
| --- | --- |
| **Вариант** | **Заданный набор ключей** |
| **34** | 2,4,6,8,10,12,13,14,15,17,18,20,21,21,24,27,28,31,35,37,39,39 |

**ПОСЛЕДОВАТЕЛЬНЫЙ ПОИСК**

1. **Сформировать алгоритмы поиска.**



1. **Провести поиск заданного ключа.**

{2, 4, 6, 8, 10, 12, 13, 14, 15, 17, 18, 20, 21, 21, 24, 27, 28, 31, 35, 37, 39, 39}

**Шаг 1**: N := 22; K := 31; i := 1;

**Шаг 2**: **K∨K1** -> 31 ≠ 2 -> i := i + 1;

**Шаг 3**: **K∨K2** -> 31 ≠ 4 -> i := i + 1;

**Шаг 4**: **K∨K3** -> 31 ≠ 6 -> i := i + 1;

**Шаг 5**: **K∨K4** -> 31 ≠ 8 -> i := i + 1;

...

**Шаг 19**: **K∨K18** -> 31 = 31 -> Искомый ключ найден под номером **31**.

**БИНАРНЫЙ ПОИСК**

1. **Сформировать алгоритмы поиска.**



1. **Провести поиск заданного ключа.**

{2, 4, 6, 8, 10, 12, 13, 14, 15, 17, 18, 20, 21, 21, 24, 27, 28, 31, 35, 37, 39, 39}

**Шаг 1**: **R** := 22; **K** := 31; **L** := 1;

**Шаг 2**: **n** := (L + R) div 2 = (1 + 22) div 2 = 11;

**Шаг 3**: **K∨K11** -> 31 > 18 -> Дальнейший поиск в правом подмножестве.

**Шаг 4**: **L** := n + 1 = 11 + 1 = 12;

**Шаг 5**: **n** := (L + R) div 2 = (12 + 22) div 2 = 17;

**Шаг 6**: **K∨K17** -> 31 > 28 -> Дальнейший поиск в правом подмножестве.

**Шаг 7**: **L** := n + 1 = 17 + 1 = 18;

**Шаг 8**: **n** := (L + R) div 2 = (18 + 22) div 2 = 20;

**Шаг 9**: **K∨K20** -> 31 < 37 -> Дальнейший поиск в левом подмножестве.

**Шаг 10**: **R** := n = 20;

**Шаг 11**: **n** := (L + R) div 2 = (18 + 20) div 2 = 19;

**Шаг 12**: **K∨K19** -> 31 < 35 -> Дальнейший поиск в левом подмножестве.

**Шаг 13**: **R** := n = 19;

**Шаг 14**: **n** := (L + R) div 2 = (18 + 19) div 2 = 18;

**Шаг 15**: **K∨K18** -> 31 = 31 -> Искомый ключ найден под номером 18.

**ФИБОНАЧЧИЕВ ПОИСК**

1. **Сформировать алгоритмы поиска.**



**Описание алгоритма:**

**Шаг 1:**

* **Введить:**
  + **K** – ключ.
  + **N** – количество элемента;
* **Установим:**
  + **i** = 1; **p** = 2; **q** = 3;

**Шаг 2:**

Сравним K и Ki.

* Если K > Ki
  + Если i < N -> Перейти на **Шаг 3.**
  + Если i > N -> Перейти на **Шаг 4.**
  + Если i = N -> **Ключ K не найден.**
* Если K < Ki -> Перейти на **Шаг 4.**
* Если K = Ki -> **Ключ** **K найден.**

**Шаг 3:**

* a = i;
* i = p + b;
* p = q + b;
* q = i + p – c;
* b = 0;
* Перейти на **Шаг 2.**

**Шаг 4:**

* b = a – 1;
* c = b;
* p = 1;
* q = 2;
* Перейти на **Шаг 3.**

1. **Провести поиск заданного ключа.**

{2, 4, 6, 8, 10, 12, 13, 14, 15, 17, 18, 20, 21, 21, 24, 27, 28, 31, 35, 37, 39, 39}

**Шаг 1.** **K∨k1** -> 31 > 2 -> сравнение продолжается.

**Шаг 2. K∨k2** -> 31 > 4 -> сравнение продолжается.

**Шаг 3. K∨k3** -> 31 > 6 -> сравнение продолжается.

**Шаг 4. K∨k5** -> 31 > 12 -> сравнение продолжается.

**Шаг 5. K∨k8** -> 31 > 14 -> сравнение продолжается.

**Шаг 6. K∨k13** -> 31 > 21 -> сравнение продолжается.

**Шаг 7. K∨k21** -> 31 < 39 -> найден интервал, в котором находится отыскиваемый ключ, т.е отыскиваемый ключ может находится в исходном множестве **между 13 и 21 позициями.**

**То есть {**21, 21, 24, 27, 28, 31, 35, 37, 39**}**

**Шаг 1.** **K∨k13** -> 31 > 21 -> сравнение продолжается.

**Шаг 2.** **K∨k14** -> 31 > 21 -> сравнение продолжается.

**Шаг 3.** **K∨k15** -> 31 > 24 -> сравнение продолжается.

**Шаг 4.** **K∨k17** -> 31 > 28 -> сравнение продолжается.

**Шаг 5.** **K∨k20** -> 31 < 37 -> найден интервал, в котором находится отыскиваемый ключ, т.е отыскиваемый ключ может находится в исходном множестве **между 17 и 20 позициями.**

**То есть {**28, 31, 35, 37**}**

**Шаг 1.** **K∨k17** -> 31 > 28 -> сравнение продолжается.

**Шаг 2.** **K∨k18** -> 31 > 31 -> Искомый ключ найден **под номером 18**.

**Вывод:**

Наименьшее количество шагов, за которые можно было найти ключ, было достигнуто при поиске по Фибоначчиев поиск (14 шагов). Поэтому он является в данном случае наилучшим. Следующий по эффективности – бинарному поиску (15 шагов). Наибольшее количество шагов было при последовательном поиске (19 шагов), поэтому его эффективность наименьшая среди остальных методов поиска.