Алгоритмы поиска.

Задача поиска n элементов записи, каждая из которых сопровождается ключом поиска и некоторой информационной частью. Информационную часть можно не рассматривать, так как чаще ищется по ключу. Требуется найти позицию ключа х. Результат поиска:

-всегда либо «нашел» либо «не нашел».   
-Может найти позицию первого вхождения,   
-последнего вхождения,   
-а может найти все вхождения элементов в наш вектор включения.

Виды поиска:

1.Линейный поиск

Const N=10;  
var i,x: integer;  
k = array [1..N] of integer;

Begin  
 i:=1;  
 while(i<=N) and (k[i]<>x) do  
 inc(i);  
 if <=N then  
 write(‘Нашла’);  
 else  
 write(‘Не нашла’);  
end;

2. Линейный поиск с барьером

Const N=10;  
var i,x: integer;  
k = array [1..N+1] of integer;

Begin  
 i:=1;  
 k[N+1]:=x; //барьер  
 while (k[i]<>x) do  
 inc(i);  
 if <=N then  
 write(‘Нашла’);  
 else  
 write(‘Не нашла’);  
end;

3. Бинарный поиск

l,u – нижняя, верхняя границы  
l := 1; u := N;

while (l<=N) do begin  
i:=(u+l)/2;  
if k[i] = x then  
begin

writeln(‘нашли’); goto 1\_end;

if k[i] < x then u:=i+1;

if k[i] > x then u:=i-1;  
end;

writeln (‘не нашли’);

1\_end: end;

4. Однородный бинарный поиск

D[1]..D[max]  
Max = [log2N]+2  
D[i] = [ (N+2i-1)/2i ], i = 1,max

1. Сформировать вектор D
2. i= D[1]; j=2;
3. if x<k[i] , if D[j] = 0 then writeln (‘Не нашли’) else begin  
   i:=i-D[j]; inc(j); doto (3); end;
4. if x>k[i] if D[j] = 0 then writeln (‘не нашли’) else begin  
   i = i+D[j]; inc(j); goto(3)
5. if x=k[i] then writeln (‘нашли’); end;

5. Интерполяционный

l=1; u=N;

while l<=N do begin  
i = l + [ ( u - l ) + ( x + k[l] ) / (k[u] – k[l]) ];

if x<k[i] then u:=i-1;

if x>k[i] then l:=i+1;

if x=k[i] then writeln (‘нашли’);

end;