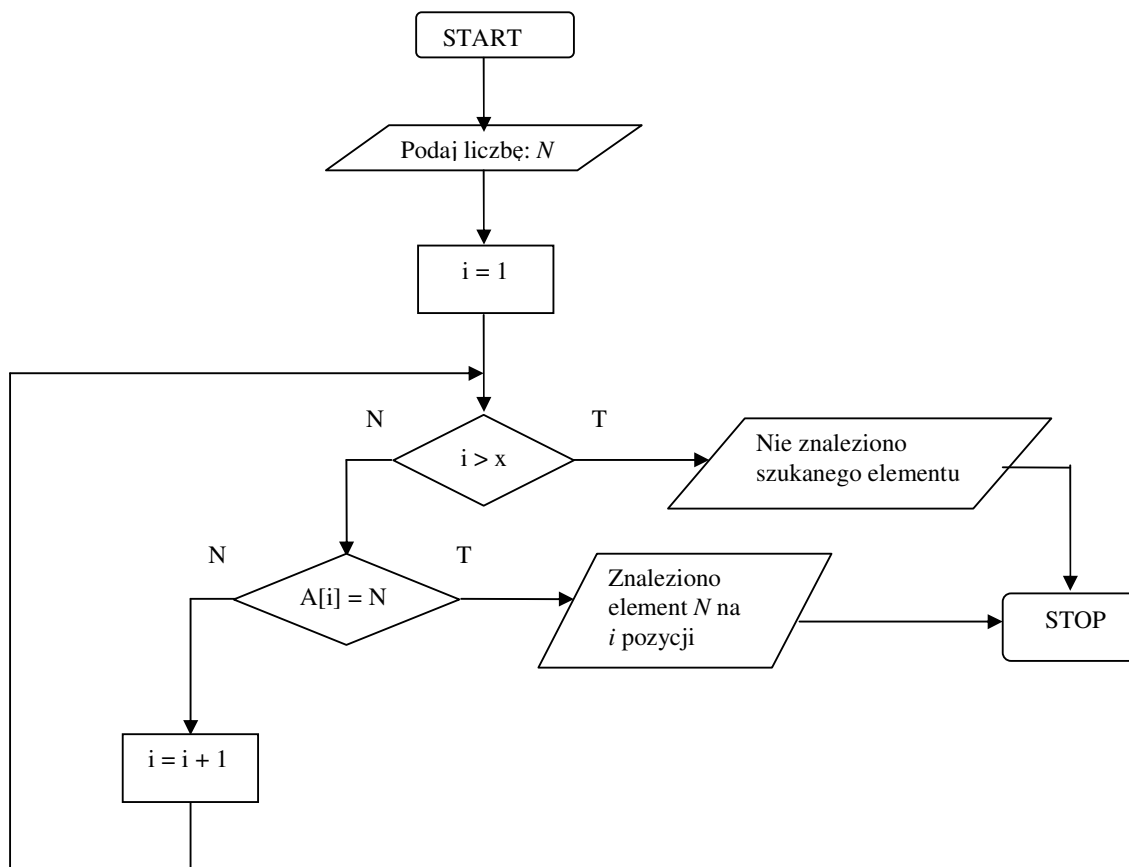


Uniwersytet Zielonogórski	Wykonali:	Grupa:	Nr ćwiczenia: 2	Ocena:
Laboratorium				
Temat ćwiczenia: Przeszukiwanie liniowe, binarne, laszowanie.		Prowadzący:	Data wyk. ćw.	Data odd. spr.

1. Przeszukiwanie liniowe.

- schemat blokowy



Tablica A jest x - elementowa

- kod źródłowy w języku Turbo Pascal

```
program prze_lin;
uses crt;
const x=20;
type Tab=array[1..x] of integer;
var A:Tab; i,k:byte; N:integer;

BEGIN
clrscr;

write('Podaj szukana liczbe: ');
readln(N);

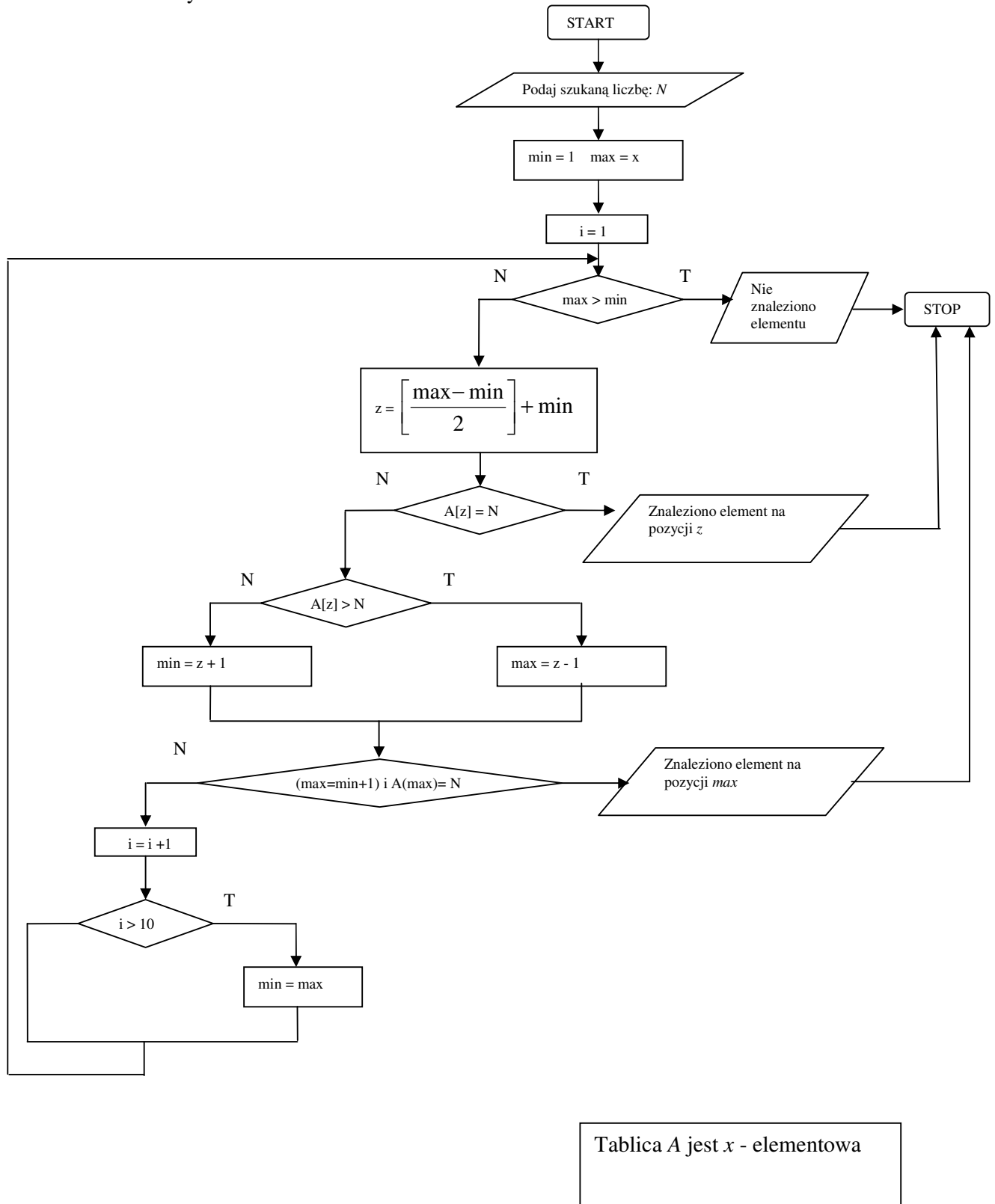
k:=0;
for i:=1 to x do begin
  if A[i]=N then begin k:=1; break; end; end;

if k=1 then write('Znaleziono element na ',i,' pozycji.')
  else write ('Nie znaleziono elementu.');
```

```
readln;
END.
```

2. Wyszukiwanie binarne.

- schemat blokowy



- kod źródłowy w języku Turbo Pascal

```
program przesz_bin;  
uses crt;  
const x=20;  
type Tab = array[1..x]of integer;  
var A:Tab; min,max,z,N,i:integer;
```

```
BEGIN
```

```
clrscr;
```

```
write('Podaj szukana liczbe: ');  
readln(N);
```

```
i:=1;  
min:=0;  
max:=x;
```

```
while max>min do begin  
z:=trunc((max-min)/2+min);
```

```
if A[z]=N then begin write('Znaleziono element na ',z,' pozycji.');
```

```
break; end  
else begin if A[z]>N then max:=z else min:=z; end;  
if (max=min+1)and(A[max]=N) then begin  
write('Znaleziono element na ',z+1,' pozycji.');
```

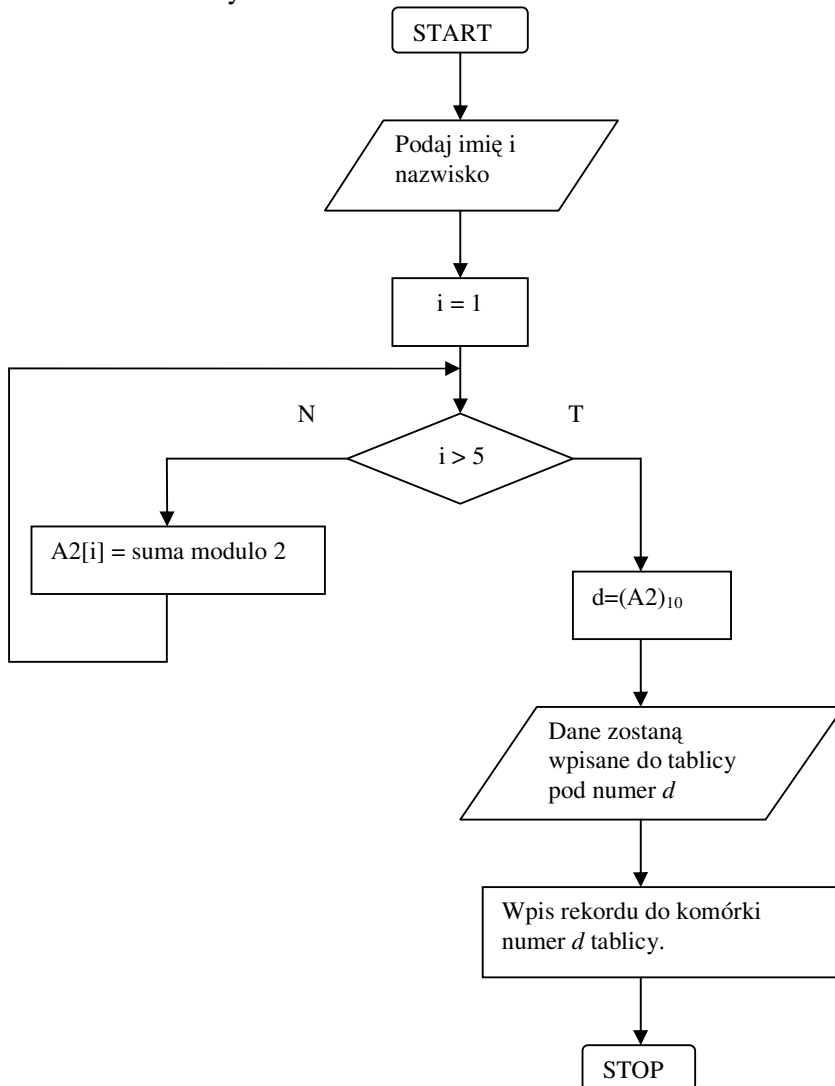
```
break; end;  
i:=i+1;  
if i>10 then min:=max; end;
```

```
if min=max then write('Nie znaleziono szukanego elementu.');
```

```
readln  
END.
```

3. Haszowanie – suma modulo 2.

- schemat blokowy



- kod źródłowy w języku Turbo Pascal

```

program sm2;
uses crt;
type Tab=array['a'..'z',1..5]of byte;
      Tab2=array[1..5] of byte;

      Dane = record
      Imie:String[5];
      Nazwisko:String[5]; end;

      Tab1=array[1..32]of Dane;
const A:Tab=((0,0,0,0,1),(0,0,0,1,0),(0,0,0,1,1),(0,0,1,0,0),(0,0,1,0,1),
            (0,0,1,1,0),(0,0,1,1,1),(0,1,0,0,0),(0,1,0,0,1),(0,1,0,1,0),
            (0,1,0,1,1),(0,1,1,0,0),(0,1,1,0,1),(0,1,1,1,0),(0,1,1,1,1),
            (1,0,0,0,0),(1,0,0,0,1),(1,0,0,1,0),(1,0,0,1,1),(1,0,1,0,0),
            (1,0,1,0,1),(1,0,1,1,0),(1,0,1,1,1),(1,1,0,0,0),(1,1,0,0,1),
            (1,1,0,1,0));

var Im,Na:String[5]; d,i:byte; A2:Tab2; R:Tab1;

BEGIN
Clrscr;
write('Podaj imie: '); readln(Im);
write('Podaj nazwisko: '); readln(Na);

for i:=1 to 5 do
A2[i]:=A[Na[1],i]xor A[Na[2],i]xor A[Na[3],i]xor A[Na[4],i]xor A[Na[5],i];

d:=A2[1]*16+A2[2]*8+A2[3]*4+A2[4]*2+A2[5]*1;

writeln('Dane zostana zapisane do tablicy pod numer ',d);

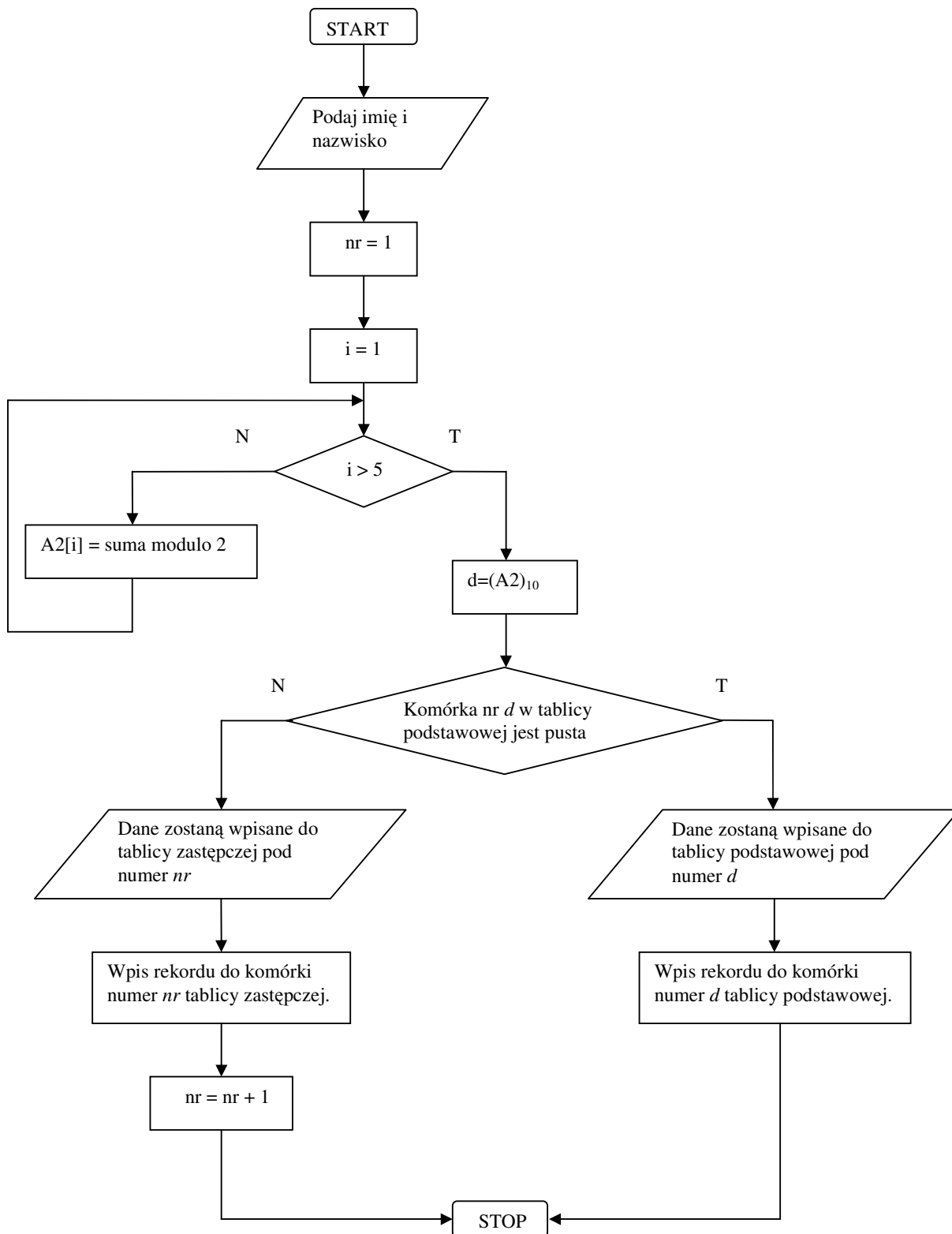
R[d].nazwisko:=Na;
R[d].imie:=Im;

readln
END.

```

4. Haszowanie - tablice.

- schemat blokowy



- kod źródłowy w języku Turbo Pascal

```

program Tablice;
uses crt;
type Tab=array['a'..'z',1..5]of byte;
     Tab2=array[1..5] of byte;
     Dane = record
         Imie:String[5];
         Nazwisko:String[5]; end;
     Tab1=array[1..32]of Dane;
const A:Tab=((0,0,0,0,1),(0,0,0,1,0),(0,0,0,1,1),(0,0,1,0,0),(0,0,1,0,1),
            (0,0,1,1,0),(0,0,1,1,1),(0,1,0,0,0),(0,1,0,0,1),(0,1,0,1,0),
            (0,1,0,1,1),(0,1,1,0,0),(0,1,1,0,1),(0,1,1,1,0),(0,1,1,1,1),
            (1,0,0,0,0),(1,0,0,0,1),(1,0,0,1,0),(1,0,0,1,1),(1,0,1,0,0),
            (1,0,1,0,1),(1,0,1,1,0),(1,0,1,1,1),(1,1,0,0,0),(1,1,0,0,1),
            (1,1,0,1,0));
var Im,Na:String[5]; d,i,nr:byte; A2:Tab2; R,R1:Tab1;

BEGIN
Clrscr;

nr:=1;

for i:=1 to 32 do R[i].nazwisko:='*';

write('Podaj imie: '); readln(Im);
write('Podaj nazwisko: '); readln(Na);

for i:=1 to 5 do
A2[i]:=A[Na[1],i]xor A[Na[2],i]xor A[Na[3],i]xor A[Na[4],i]xor A[Na[5],i];

d:=A2[1]*16+A2[2]*8+A2[3]*4+A2[4]*2+A2[5]*1;

If R[d].nazwisko='*' then begin
writeln('Dane zostana zapisane do tablicy podstawowej pod numer ',d);
R[d].nazwisko:=Na;
R[d].imie:=Im;           end else begin
writeln('Dane zostana zapisane do tablicy zastepczej pod numer ',nr);
R1[nr].nazwisko:=Na;
R1[nr].imie:=Im;
nr:=nr+1;                end;

readln
END.

```

Wnioski:

Przeszukiwanie liniowe jest bardzo prostym sposobem przeszukania tablicy. Polega ono na przeglądaniu tablicy element po elemencie i porównywaniu za każdym razem wartości danego elementu z wartością poszukiwaną. Jego zaletą jest to, że przeszukiwana tablica nie musi być uporządkowana, wadą natomiast jest to, że jeśli szukany element znajduje się na końcu tablicy, to przeszukane zostaną wszystkie poprzednie komórki tablicy, co wiąże się z czasem operacji.

Wyszukiwanie binarne można stosować tylko na tablicach uporządkowanych. Zaczyna się od środkowego elementu i porównuje z szukaną wartością. Jeżeli jest mu równa, to przeszukiwanie zostaje zakończone. Jeżeli jest on większy niż szukana wartość, to odrzuca się dolną połowę tablicy i ogranicza się poszukiwanie do elementów powyżej środka. Jeżeli natomiast środkowy element jest mniejszy od szukanej wartości, to odrzuca się górną połowę i kontynuuje się przeszukiwanie jedynie dolnej części tablicy. Zaletą takiego wyszukiwania jest mniejsza ilość operacji potrzebna do przeszukania całej tablicy niż w przypadku przeszukiwania liniowego.

Haszowanie jest już bardziej złożoną operacją. Suma modulo 2 koduje dany wyraz (w naszym przypadku nazwisko) na kod dwójkowy, następnie na kod dziesiętny i pod ten numer komórki do tablicy zapisywany jest cały rekord.

Haszowanie za pomocą tablic polega na stworzeniu dwóch tablic – podstawowej i zastępczej. Tu również nazwisko jest kodowane na kod dwójkowy, następnie na kod dziesiętny i jeśli ten numer komórki w tablicy podstawowej nie jest zajęty, to zapisywany jest tam cały rekord. Jeśli komórka tablicy podstawowej o tym numerze jest zajęta, to rekord ten jest wpisywany do pierwszego wolnego pola tablicy zastępczej.