Tehnica Greedy

Realizat de Andrei Habasescu
PROFESOR : GUTU MARIA

Contents

Descrierea Tehnicii Greedy

Metoda de programare **Greedy** se aplică problemelor de optimizare. Aceasta metoda constă în faptul că se construieşte solutia optimă pas cu pas, la fiecare pas fiind selectat în solutie elementul care pare „**cel mai bun/cel mai optim**" la momentul respectiv, în speranta că această alegere locală va conduce la **optimul globa**l.

Algoritmii Greedy sunt foarte eficienti, dar nu conduc în mod necesar la o solutie optimă. Şi nici nu este posibilă formularea unui criteriu general conform căruia să putem stabili exact dacă metoda Greedy rezolvă sau nu o anumită problemă de optimizare. Din acest motiv, orice algoritm Greedy trebuie însotit de o demonstratie a corectitudinii sale . Demonstratia faptului că o anumită problemă are proprietatea alegerii Greedy se face de obicei prin inductie matematică [1].

Exemplu Teoretic

Metoda **Greedy** se aplică problemelor pentru care se dă o **mulţime A** cu **n** elemente şi pentru care trebuie determinată o **submulţime** a sa, **S** cu **m** elemente, care îndeplinesc anumite condiţii, numite si **conditii de optim**. Algoritmul in limbaj natural al metodei de programare Greedy are urmatoarea structura [1]:

Algoritm Greedy:

- se dă o mulţime **A**
- se cere o submulţime **S** din multimea **A** care sa :
  îndeplinească anumite condiţii interne (să fie acceptabilă)
  să fie optimală (să realizeze un maxim sau un minim).

Principiul metodei Greedy:

- se **iniţializează** mulţimea soluţiilor S cu mulţimea vidă, S=Ø
- la fiecare pas se alege un anumit element **x∈A** (cel mai promiţător element la momentul respectiv) care poate conduce la o soluţie optimă
- se verifică dacă elementul ales poate fi adăugat la mulţimea soluţiilor:
- **dacă** da **atunci** :
  va fi adăugat şi mulţimea soluţiilor devine S=S∪{x} - un element introdus în mulţimea **S** nu va mai putea fi eliminat
- **altfel** :
  el nu se mai testează ulterior

procedeul continuă, până când au fost determinate toate elementele din mulţimea soluţiilor.

Avantaje & Dezavantaje

+

- Timpul de rezolvare este mic, algorimul Greedy fiind unul polinomial
- Dacă condiția si formularea programului este corectă, algoritmul Greedy va găsi mereu o soluție

-

- Nu toate problemele au o condiție bine definită, astfel algoritmul Greedy nu este aplicabil acestora.
- Deşi algoritmul Greedy va găsi o soluție in cazul că condiția si formularea programului sunt valide, această soluție poate să nu fie optimală problemei.

Exemple Practice

1) Exemplul clasic al problemelor rezolvate prin algoritmul **Greedy** (calcularea numarului de monede de diferite valori necesare pentru returnarea restului)

```
Program rest;
type coins = array[1..5] of integer;
var
x,i:integer;
a,b:coins;
begin
  a[1]:=50; a[2]:=25; a[3]:=10; a[4]:=5; a[5]:=1;

  write('Introduceti numarul de banuti (rest) : ');
  readln(x);

  i:=1;
  while x>0 do begin
    if x-a[i]>=0 then begin
      x:=x-a[i];
      inc(b[i]);
    end else begin
      inc(i);
```

```
    end;
  end;

  writeln();
  writeln('Pentru a intoarce acest rest aveti nevoie de urmatorul set de
banuti : ');
  writeln();
  for i:=1 to 5 do writeln(a[i],'x ',b[i]);
end.
```

2) Împărțirea clasei in 2 părți dupa sex aplicând algoritmul **Greedy**

```
Program impartirea_clasei_dupa_sex;
type
data = record
  name : string;
  gender : char;
end;

tab = array[1..100] of data;
var
a,b:tab;
i,n,n1:integer;
x:char;


function checkFemale(var a:tab):boolean;
  var i:integer;
  begin
    checkFemale:=False;
    i:=1;
    while (a[i].gender<>'F') and (i<=n) do inc(i);
    if (i<=n) and (a[i].gender='F') then checkFemale:=True;
  end;

procedure extractFemale(var a,b:tab; var x:integer);
  var i:integer;
  begin
    i:=1;
    while (i<=n) and (a[i].gender<>'F') do inc(i);
    inc(x);
    b[x].gender:=a[i].gender;
    a[i].gender:='-';
    b[x].name:=a[i].name;
    a[i].name:='N/A'
  end;

begin
  i:=0;
  while x<>'N' do begin
    inc(i);
    writeln(i,': ');
    write(' nume : ');
```

```
    readln(a[i].name,x);
    write(' sex[M/F] : ');
    readln(a[i].gender);
    writeln('continue list creation?');
    writeln('----------Y/N----------');
    readln(x);
  end;
  n:=i;
  while checkFemale(a)=true do extractFemale(a,b,n1);

  writeln('----- Lista Fetelor -----');
  for i:=1 to n1 do writeln(b[i].name);
  writeln('------Lista Initiala-----');
  for i:=1 to n do writeln(a[i].name);
end.
```

3) Sortarea unei liste de muzică dupa criteriul genului de muzică/al artistului, aplicând algoritmul **Greedy**

```
Program muzica;
type
data = record
  name : string;
  artist : string;
  genre : string;
end;

tab = array[1..100] of data;
var
a,a_copy,b:tab;
i,n,n1:integer;
input:string;
x:char;

{----------------------------------------}
function checkGenre(var a:tab):boolean;
  var i:integer;
  begin
    checkGenre:=False;
    i:=1;
    while (a[i].genre<>input) and (i<=n) do inc(i);
    if (i<=n) and (a[i].genre=input) then checkGenre:=True;
  end;

procedure includeItem_Genre(var a,b:tab; var x:integer);
  var i:integer;
  begin
    i:=1;
    while (i<n) and (a[i].genre<>input) do inc(i);
    inc(x);
    b[x].artist:=a[i].artist;
    a[i].artist:='N/A';
    b[x].genre:=a[i].genre;
```

```
    a[i].genre:='N/A';
    b[x].name:=a[i].name;
    a[i].name:='N/A';
  end;

{----------------------------------------}
procedure recovery();
  var i:integer;
  begin
    a_copy:=a;
    for i:=1 to n1 do begin
      b[i].artist:='';
      b[i].genre:='';
      b[i].name:='';
    end;
    n1:=0;
  end;
{----------------------------------------}
function checkArtist(var a:tab):boolean;
  var i:integer;
  begin
    checkArtist:=False;
    i:=1;
    while (a[i].artist<>input) and (i<=n) do inc(i);
    if (i<=n) and (a[i].artist=input) then checkArtist:=True;
  end;

procedure includeItem_Artist(var a,b:tab; var x:integer);
  var i:integer;
  begin
    i:=1;
    while (i<n) and (a[i].artist<>input) do inc(i);
    inc(x);
    b[x].artist:=a[i].artist;
    a[i].artist:='N/A';
    b[x].genre:=a[i].genre;
    a[i].genre:='N/A';
    b[x].name:=a[i].name;
    a[i].name:='N/A';
  end;

procedure filterArtist();
var i:integer;
  begin
    write('Numele artistului : '); readln(input);
    if checkArtist(a_copy) then includeItem_Artist(a_copy,b,n1);
  end;

procedure filterGenre();
  var i:integer;
  begin
    write('Denumirea genului de muzica : '); readln(input);
    if (checkGenre(a_copy) = true) then includeItem_Genre(a_copy,b,n1);
  end;

begin
  i:=0;
```

```
  while x<>'N' do begin
    inc(i);
    writeln(i,': ');
    write(' Arist : ');
    readln(a[i].artist);
    write(' Genre  : ');
    readln(a[i].genre);
    write(' Name  : ');
    readln(a[i].name);
    writeln('continue list creation?');
    writeln('----------Y/N----------');
    readln(x);
  end;
  n:=i;

  a_copy:=a;

  while x<>'X' do begin
    writeln();
    writeln();
    writeln('G - Filtrare dupa gen de muzica');
    writeln('A - Filtrare dupa artist');
    writeln('L - Afisarea listei originale');
    writeln('N - Adaugare elemente');
    writeln('X - Iesire din program');
    readln(x);

    if x = 'G' then begin
      filterGenre();
      for i:=1 to n1 do writeln(' ',b[i].artist,' - ',b[i].name);
      recovery();
    end else
      if x = 'A' then begin
        filterArtist();
        for i:=1 to n1 do writeln(' ',b[i].name,' - ',b[i].genre);
        recovery();
      end else
        if x = 'L' then for i:=1 to n do writeln('  ',i,'# ',a[i].name,' -
',a[i].artist,' : ',a[i].genre)
        else if x = 'N' then begin
            x:=' ';
            i:=n;
            while x<>'N' do begin
              writeln();
              inc(i);
              writeln(i,': ');
              write(' Arist : ');
              readln(a[i].artist);
              write(' Genre  : ');
              readln(a[i].genre);
              write(' Name  : ');
              readln(a[i].name);
              writeln('continue list creation?');
              writeln('----------Y/N----------');
              readln(x);
            end;
            n:=i;
```

```
    end;
  end;
end.
```

4) Extragerea numerelor pozitive/negative, aplicând algoritmul **Greedy**

```
Program sortare_numere;
type tab = array[1..100] of integer;
var i,j,n:integer;
    a,b:tab;
    x:char;

  procedure pos(var a:tab; var j:integer);
    var i:integer;
    begin
      for i:=1 to n do if a[i]>0 then begin
        inc(j);
        b[j]:=a[i];
      end;
    end;

   procedure neg(var a:tab; var j:integer);
    var i:integer;
    begin
      for i:=1 to n do if a[i]<0 then begin
        inc(j);
        b[j]:=a[i];
      end;
    end;

   procedure reset();
    var i:integer;
    begin
      for i:=1 to j do b[i]:=0;
      j:=0;
    end;

begin
  write('numarul de elemente al tabelului : '); readln(n);

  writeln();
  for i:=1 to n do begin
    write(i,'# : '); readln(a[i]);
  end;

  while x<>'X' do begin
    writeln('-----------------------------------');
    writeln('P - Extragerea elementelor pozitive');
    writeln('N - Extragerea elementelor negative');
    writeln('X - Iesire din program');
    writeln();
    readln(x);
```

```
    if x = 'P' then begin
      pos(a,j);
      for i:=1 to j do writeln(i,'# ',b[i]);
      reset();
    end else
      if x = 'N' then begin
        neg(a,j);
        for i:=1 to j do writeln(i,'# ',b[i]);
        reset();
      end;
  end;

end.
```

5) Sortarea listei de vehicule după gara cărei acestea le aparțin, aplicând algoritmul **Greedy**

```
Program gari_Auto;
type
data = record
  ID : string;
  terminal : string;
end;

tab = array[1..100] of data;
var
a,b:tab;
i,n,n1:integer;
input:string;
x:char;

function checkTerminal(var a:tab):boolean;
  var i:integer;
  begin
    checkTerminal:=False;
    i:=1;
    while (a[i].terminal<>input) and (i<=n) do inc(i);
    if (i<=n) and (a[i].terminal=input) then checkTerminal:=True;
  end;

procedure extractVehicle(var a,b:tab; var x:integer);
  var i:integer;
  begin
    i:=1;
    while (i<=n) and (a[i].terminal<>input) do inc(i);
    inc(x);
    b[x].terminal:=a[i].terminal;
    a[i].terminal:='-';
    b[x].ID:=a[i].ID;
    a[i].ID:='N/A'
  end;

begin
```

```
  i:=0;
  while x<>'N' do begin
    inc(i);
    writeln(i,': ');
    write(' Numerele Inmatriculare : ');
    readln(a[i].ID,x);
    write(' Gara Carei Apratine Vehiculul  : ');
    readln(a[i].terminal);
    writeln('continue list creation?');
    writeln('----------Y/N----------');
    readln(x);
  end;
  n:=i;

  write('Introduceti numele garii a carei vehicule doriti sa fie afisate :
');
  readln(input);

  while checkTerminal(a)=true do extractVehicle(a,b,n1);

  writeln;
  writeln;
  writeln;
  writeln;
  writeln('----- Lista Vehiculelor al garii ',input,'-----');
  for i:=1 to n1 do writeln(b[i].ID);
end.
```

Concluzii

Metoda Greedy este foarte eficientă atunci cînd dorim să aflăm rezultatul optim în cît mai scurt timp posibil, deoarece algoritmii sunt polinomiali.

Defectul acestei metode este că aceasta poate fi aplicată numai atunci cînd din enunțul problemei poate fi dedusă regula care asigură selecția directă a elementelor necesare din mulțimea data [2].

Bibliografie

1. https://sites.google.com/site/eildegez/home/clasa-xi/prezentarea-metodei-greedy
2. http://caterinamacovenco.blogspot.com/p/metoda-greedy.html
3. https://www.researchgate.net/figure/Advantages-and-drawbacks-of-genetic-and-greedy-algorithms_tbl2_221472685
4. https://en.wikipedia.org/wiki/Greedy_algorithm