Žiogas

Žinome, kad virvutė yra *s* sprindžių ilgio. Tarkime, kad joje kas sprindį bei abiejuose galuose užrišta po mazgą. Mazgai sunumeruoti nuo 0 iki *s*. Susidarysime lentelę, kurioje žymėsimės kiek mažiausiai žingsnių reikia padaryti, norint pasiekti atitinkamą mazgą. Jei mazgas nepasiekiamas, lentelės elemente įrašytas -1.

Pavyzdžiui, pradiniai duomenys yra tokie: $s = 11$, $c = 5$, $a = 3$, $b = 1$. Sudarome lentelę iš 12 elementų:

Mazgo nr.	0	1	2	3	4	5	6	7	8	9	10	11
Per kiek žingsnių jis pasiekiamas	**0**	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Pradžioje žiogas stovi ant virvutės krašto, taigi nulinis mazgas yra pasiektas. Peržiūrime lentelę, randame visus mazgus, kuriuose galima pasiekti per nulį žingsnių (šiuo atveju bus vienas toks mazgas) ir nuo jo žiogas šoka *a* sprindžių pirmyn. Atgal šokti žiogas negali, nes nukris nuo virvutės.

Mazgo nr.	0	1	2	3	4	5	6	7	8	9	10	11
Per kiek žingsnių jis pasiekiamas	**0**	-1	-1	**1**	-1	-1	-1	-1	-1	-1	-1	-1

Dabar peržiūrime visus mazgus, randame tuos, kuriuos žiogas gali pasiekti padaręs tik vieną šuolį ir nuo jų žiogas šoka pirmyn ir atgal.

Mazgo nr.	0	1	2	3	4	5	6	7	8	9	10	11
Per kiek žingsnių jis pasiekiamas	**0**	-1	**2**	**1**	-1	-1	**2**	-1	-1	-1	-1	-1

Taip pildome lentelę, kol pasiekiame mazgą, nutolusį nuo pradžios per *c* sprindžių arba žiogas nebegalės pasiekti nei vieno mazgo, kuriame jo nebūta. Baigus darbą su šiais pradiniais duomenimis, lentelė turėtų atrodyti taip:

Mazgo nr.	0	1	2	3	4	5	6	7	8	9	10	11
Per kiek žingsnių jis pasiekiamas	**0**	**3**	**2**	**1**	-1	**3**	**2**	-1	-1	**3**	-1	-1

```
const MAX = 1000; { maksimalus virvutės ilgis }
type lentelė = array [0..MAX] of integer;

function šokti (s, a, b, c: integer): integer;
  var l: lentelė; { žymėsime pasiektus mazgus }
      i, žingsniai: integer;
      galima: boolean; { ar galima pasiekti naujas vietas virvutėje }
begin
  { nei vienoje virvutės vietoje žiogas dar nebuvo }
  for i := 1 to s do
    l[i] := -1;
  l[0] := 0;   { žiogas stovi ant virvutės galo }
  žingsniai := 0; { dar nepadarė nei vieno šuolio }
  galima := true;
  while galima and (l[c] = -1) do
    begin
      galima := false;
      { šoks į priekį }
      for i := 0 to s-a do
        if (l[i] = žingsniai) and (l[i+a] = -1)
            then begin
                   galima := true;
                   l[i+a] := žingsniai + 1
                 end;
      { šoks atgal }
      for i := b to s do
        if (l[i] = žingsniai) and (l[i-b] = -1)
           then begin
                  galima := true;
                  l[i-b] := žingsniai + 1
                end;
      žingsniai := žingsniai + 1
    end;
  šokti := l[c];
end; { šokti }
```