Soluţie pitici – Emanuela Cerchez

Să notăm $HTi=Hi+Li$ (înălţimea totală a piticului i).

Mai întâi să analizăm o situaţie simplă: pot ieşi toţi piticii din groapă? Da, dacă şi numai dacă există un pitic i astfel încât $HT_i>=D$ (el ar fi ultimul pitic), iar ceilalţi $N-1$ pitici pot ieşi toţi dintr-o groapă cu înălţimea $D-H_i$.
Intuim că piticii cu HT maxim trebuie să stea la bază (deci sortăm piticii descrescător după HT).

Să notăm cu $sum=H_1+H_2+...+H_N$

Vom utiliza programarea dinamică.
Fie $p[i][j]$=înălţimea minimă a unui turn format din i pitici care pot ieşi, cei i pitici fiind aleşi dintre piticii 1, 2, ..., j
$p[0][0]=0$
$p[i][j]=INF$ dacă $j>i$
La baza turnului vor fi plasati piticii care nu pot ieşi, înălţimea turnului format din aceştia fiind $sum-p[i][j]$.

```
p[0]=0;
//p[i]=inaltimea minima a unui turn format din i pitici care pot iesi
c=0;   //numarul de pitici care pot iesi
for (j=1; j<=n; j++) //consider piticii 1, 2, ..., j
     {
      for (i=c; i>=0; i--) //pot iesi i pitici
           //p[i]=inaltimea turnului format din cei i pitici care ies
           //la baza turnul are inaltimea sum-p[i]
            if (sum - p[i] + P[j].L >= D)//piticul j poate iesi
               { //la ce i pitici care pot iesi se adauga piticul j
                if (p[i+1] > p[i] + P[j].h)
                    p[i+1]=p[i] + P[j].h;
               }
     if (p[c+1] !=INF) c++;
    }
}
```

Astfel toate valorile $p[i,j]$ sunt calculate în $O(NxN)$ cu $O(N)$ memorie.
Soluţia problemei este valoarea j maximă pentru care $p[N][j]$ nu este INF.

Soluţie greedy – Mugurel Ionuţ Andreica

Vom sorta piticii crescător după înălţimea lor, obţinand o ordonare pH(1), pH(2), ..., pH(N), astfel incat $H_{pH(1)} \leq H_{pH(2)} \leq ... \leq H_{pH(N)}$. Ii vom sorta, de asemenea, crescător după criteriul H_i+L_i. Vom parcurge piticii in ordinea crescătoare a înălţimii lor şi vom încerca sa adăugăm piticul curent la mulţimea piticilor ce vor ieşi din groapă. Justificarea este evidentă – piticii cu înălţime mai mică contribuie mai puţin la ieşirea celorlalţi pitici din groapă, deci nu are rost să îi păstrăm pe ei în groapă în detrimentul altor pitici cu înălţime mai mare. Totuşi, piticii nu vor fi scoşi din groapă în ordinea înălţimii lor, deoarece este

posibil ca un pitic mic să poată contribui la ieşirea din groapă a unor pitici cu înălţime mai mare, după care acesta poate ieşi singur (datorită lungimii mainilor sale). Cu aceste observaţii, algoritmul greedy este urmatorul:

```
S={} // S=mulţimea piticilor ce vor ieşi din groapă
pentru i=1 la N execută
 dacă (verifică(S U {pH(i)})=ok) atunci
   S = S U {pH(i)}
```

Funcţia cea mai importantă în acest algoritm este funcţia **verifică**, care determină dacă toţi piticii dintr-o submulţime W dată ca parametru pot ieşi din groapă. Verificarea se bazează pe următoarea observaţie: dacă avem stabilit setul de pitici care vrem să iasă din groapă, atunci aceştia vor ieşi în ordine crescătoare a criteriului H_i+L_i. Vom sorta toţi piticii din submulţimea W în funcţie de acest criteriu. Această sortare va lua un timp liniar de fiecare dată, deoarece ne vom folosi de sortarea iniţială a piticilor după H_i+L_i.

```
verifică(W)
  sortează piticii din W crescător după Hi+Li în ordinea pHL(1), pHL(2), ..., pHL(|W|)
  fie SH = suma înălţimilor piticilor care nu se află in W
  fie WSH = suma înălţimilor piticilor pHL(1), pHL(2), ..., pHL(|W|)
  pentru i=1 la |W| execută
    dacă (SH + WSH + LpHL(i) < înălţimea gropii) atunci
      întoarce not ok
    WSH = WSH – HpHL(i)
  întoarce ok
```

Soluţii mai puţin eficiente:
Programare dinamică în care se foloseşte ca parametru adâncimea gropii.