Dinamičko programiranje

Zadaci

1. Aleksa uopše ne voli da dokoličari i zbog toga stalno smišlja nove igre. Jedna od njegovih omiljenih igara je *precrtavanje*. Igra se sastoji u tome da se na papiru napiše niz brojeva. Igrač ima pravo da izbriše jedan element iz tog niza (recimo a_k) i tada se iz niza brišu i elementi a_{k-1} i a_{k+1}. Ovaj potez igraču donosi a_k poena. Aleksa je perfekcionista pa Vas je zamolio da procenite koji je najveći broj poena koje može da dobije .

   Rešenje: Ovaj zadatak predstavlja varijantu problema maksimalne sume nesusednih elementa niza.

2. Data je matrica a dimenzije $m \times n$ popunjena celim brojevima. Sa svakog polja u matrici je dozvoljeno preći samo na polje ispod ili na polje desno od tog polja. Potrebno je izabrati put od gornjeg levog polja do donjeg desnog polja tako da zbir brojeva u poljima preko kojih se ide bude maksimalan.

   Rešenje :
   Očigledno je da generisanje svih mogućih puteva nije pravi pristup.
   Pretpostavimo da smo uspeli da pronađemo put od polja $(1,1)$ do polja (m, n) koji ima najveću moguću sumu. Označimo polja kroz koja prolazimo na tom putu redom brojevima $1,2,3\ldots(m+n-1)$ (dužina puta od prvog polja do polja (m, n) je $m+n-1$ jer je dozvoljeno samo kretanje na dole i u desno) a sam put sa P. Lako se može uočiti da tada svaki deo ovog puta ima maksimalnu sumu. Pretpostavimo suprotno – ako od polja $pocetak$ do polja $kraj$ $(kraj, pocetak \in \{1,2,..(m+n-1)\}, pocetak \leq kraj)$ postoji neki drugi put koji ima veću sumu nego put iz P, tada put P ne bi imao maksimalnu sumu .

   Uvedimo $d[i, j]$ da označimo maksimalnu sumu na putu od polja $(1,1)$ do polja (i, j). Rešenje zadatka je $d[m, n]$. S obzirom na to da se do polja (i, j) može doći samo preko polja $(i-1, j)$ ili polja $(i, j-1)$ matricu d možemo popuniti rekurzivno :

   $$d[i, j] = \max\{d[i-1][j], d[i], [j-1]\} + a[i][j]$$

   Elemente prve vrste i prve kolone matrice d možemo pupuniti na samo jedan način jer su jedino dozvoljeni prelasci na polje ispod odnosno polje desno od trenutnog polja. Dakle, elementi prve vrste, odnosno prve kolone matrice d se mogu uzeti kao baza rekurzije i popunjavaju se na sledeći način :

   $$d[1,1] = a[1][1]$$
   $$d[i][1] = d[i-1][1] + a[i][1], i \geq 2$$
   $$d[1][i] = d[1][i-1] + a[1][i-1], i \geq 2$$

   Da bi se rekonstruisao put (koji ima maksimalna zbir) , uvodi se još jedna matrica

   $$put[i][j] - polje\ sa\ kojeg\ se\ prešlo\ na\ polje\ (i, j)\ tako\ da\ je\ suma\ maksimalna$$

   Dakle, $put[i][j]$ može da sadrži samo jedno od polja $(i-1, j)$ i $(i, j-1)$. Pri popunjavanju matrice d, popunjava se I matrica put.

```
Input: Matrica a celih brojeva dimenzije n £ m
Output: path - niz elemenata matrice koji predstavlja traženi put
d[1][1] = a[1][1];
put[1][1] = (0; 0);
for j = 2 to m do
    d[1][j] = d[1][j - 1] + a[1][j];
    put[1][j] = (1; j - 1);
end
for i = 2 to n do
    d[i][1] = d[i - 1][1] + a[i][1];
    put[i][1] = (i - 1; 1);
end
for i = 2 to n do
for j = 2 to m do
    if d[i - 1][j] > d[i][j - 1] then
          d[i][j] = d[i - 1][j] + a[i][j];
          put[i][j] = (i - 1; j);
    else
          d[i][j] = d[i][j - 1] + a[i][j];
          put[i][j] = (i; j - 1);
    end
end
end
path = [];
trenPolje = (n;m);
while (trenPolje <> (0; 0)) do
    add trenPolje in path;
    trenPolje = put[trenPolje_ikoordniata][trenPolje_jkoordinata];
end
return path;
```

3. Data je binarna matrica dimenzija $n \times n$. Naći dimenziju najveće kvadratne matrice koja je sastavljena samo od nula.

Rešenje :

Svaka kvadratna matrica se može potpuno opisati ako se zada jedno teme (npr. donje desno) i dužina ivice. Za svako polje matrice pokušavamo da pronađemo najveću kvadratnu matricu sastavljenu samo od nula kojoj je to polje donji desno ugao.

Uvedimo pomoćnu matricu d, tako da $d[i][j]$ predstavlja dužinu najveće kvadratne matrice kojoj je polje (i, j) donji desni ugao. Matricu *d* možemo popuniti na sledeći način :

$$d[i][j] = \begin{cases} 0 & , a[i][j] = 1 \\ 1 + \min\{d[i-1][j], d[i][j-1], d[i-1][j-1], & a[i][j] = 0 \end{cases}$$

Prvu vrstu i prvu kolonu matrice d možemo jednostavno da popunimo na osnovu matrice a, dok ostale elemente popunjavamo na osnovu gore napisane formule. Rešenje je maksimum matrice d.