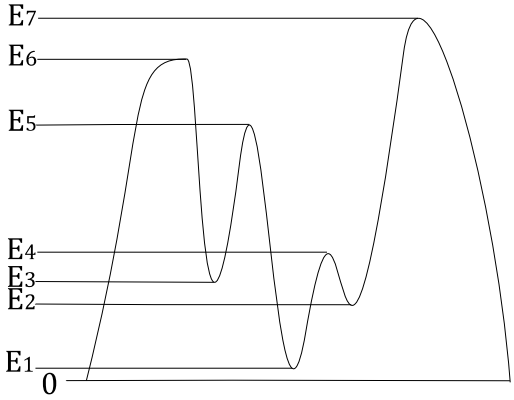
Peaks (bergstoppar)

En bergsbestigare som bor på en bergig ö i havet har klättrat upp på en topp men vill nu nå en annan, högre belägen topp.

Mer precist, varje punkt på ön har en positiv *höjd* (alltid räknad relativt havsnivån) och om den nådda toppen har höjden E_i, så är bergsbestigarens mål att nå någon topp med höjden E_j ($E_j>E_i$). Eftersom han är på en topp finns ingen omedelbar väg uppåt – för att komma till en högre punkt måste han först gå nedåt till någon lägre höjdnivå. Vägen ner är aldrig särskilt rolig, så bergsbestigaren vill maximera höjden för den lägsta punkten på vägen från den nuvarande toppen till valfri högre topp.



Exempelvis, om öns profil ser ut som i figuren och bergsbestigaren är på den topp som har höjden E_4, så finns det tre toppar med högre höjd (E_5, E_6 och E_7), men vägen med högst belägen lägstapunkt är vägen till toppen med höjden E_7 - för att nå dit måste han bara gå ner till nivån E_2 (i de övriga fallen måste han gå ner ända till E_1). Om han istället skulle starta från E_5, så skulle den högsta lägstanivån vara E_3 (vägen till E_6), medan om han startar från E_6 blir den E_1.

Kartan över ön är en tvådimensionell rektangulär tabell innehållande N×M rutor och beskriver höjden över havet för motsvarande rutor på ön. Två rutor är *angränsande* om de har en gemensam punkt. Alltså gränsar varje ruta till åtta andra (utom om den ligger vid öns kant). En väg är en serie rutor där varje ruta gränsar till den föregående. En platå är en grupp med en eller flera rutor som har samma höjd, och där varje par av rutor är sammankopplade av en väg som bara går inom gruppen. Om två angränsande rutor har samma höjd så tillhör de samma platå. En topp är en platå vars rutor inte gränsar till några rutor med högre höjd.

Skriv ett program som hittar alla toppar på ön och för var och en av dem räknar ut höjden för den högsta möjliga lägstapunkten på någon väg till en högre topp. Från en topp för vilken det inte finns någon högre topp på ön måste bergsbestigaren lämna ön för att leta efter högre toppar, så lägsta punkten kommer vara havsnivån (höjden 0).

Input data

På första raden i **peaks.in** ges heltalen N och M ($N,M\leq 2000$, $N\times M\leq 10^5$), längden respektive bredden på kartan. På de följande N raderna ges en karta över ön. Var och en av dessa rader innehåller M heltal E_{ij}($1\leq E_{ij}\leq 10^6$).

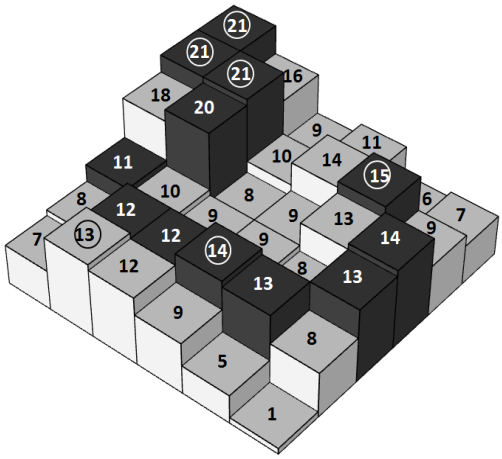
Output data

På första raden i textfilen **peaks.out** ska programmet skriva heltalet P, antalet toppar på ön. På var och en av de följande P raderna ska programmet skriva två heltal: höjden på toppen och höjden på den högsta möjliga lägstapunkten på någon väg som leder till en högre topp. Topparna ska listas i sjunkande höjdordning; om flera toppar har samma

höjd så ska dessa sorteras i sjunkande ordning med avseende på lägstapunkten (det andra utskrivna talet).

Example 1

Input data (file `peaks.in`)	Output data (file `peaks.out`)	Comment:

`6 6`
`21 16 9 11 6 7`
`21 21 10 14 15 9`
`18 20 8 9 13 14`
`11 10 9 9 8 13`
`8 12 12 14 13 8`
`7 13 12 9 5 1`
	`4`
	`21 0`
	`15 11`
	`14 13`
	`13 12`
		
		Alla topprutor är markerade med cirklar. En av de möjliga vägarna från toppen med höjd 15 visas.

Example 2

Input data (file `peaks.in`)	Output data (file `peaks.out`)

`5 3`
`16 14 16`
`14 14 15`
`12 17 16`
`12 13 10`
`16 11 16`
	`5`
	`17 0`
	`16 15`
	`16 14`
	`16 13`
	`16 13`

Grading

Testfall med N≤2 eller M≤2 är värda 15 poäng.
Testfall med P≤500 är värda 50 poäng.
Testfall med P≤5000 är värda 80 poäng.