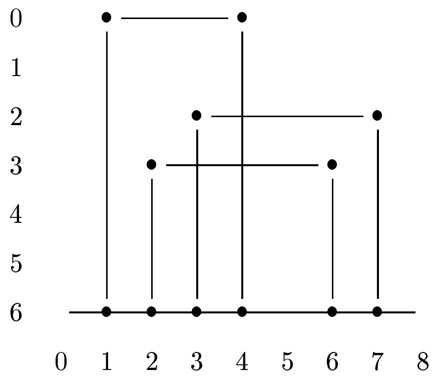
# SkyLine

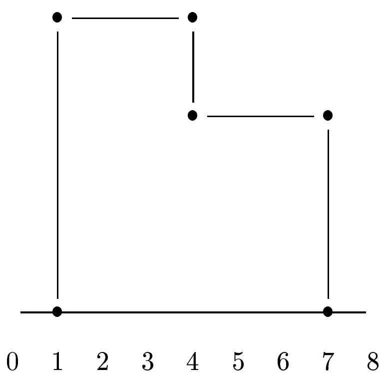
Oblig.2

## 1 Problembeskrivelse

Gitt plassering av $n$ rektangulare bygninger i en to-dimensjonal by, gi en algoritme som beregner "SkyLine" av disse ved å elimiere skjulte linjer. Tegning viser et eksempel på 'Input' og tilsvarende 'Output':



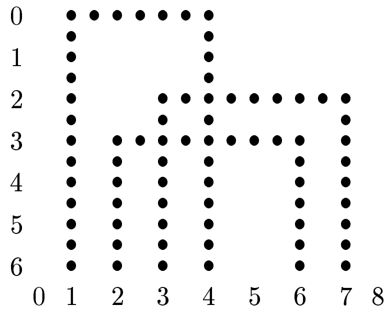

Input




Output

## 2 Matrise input

Typisk, vil data til et slikt program komme fra skjermen eller, mer generelt, fra et grafisk objekt som er en $x \times y$ matrise av *pixels*. Hvert felt i matrisen inneholder en helltalsverdi som identifiserer fargen. Tegningen 'Input' kan tenkes å være gitt på følgende måte (numerering tilsvarer forrige tegning men her er det selvsagt finere oppløsning):



Pixel-matrise

Vi har nå en slik matrise med heltall som input – tallet 0 i posisjon $x : y$ indikerer en tom pixel (med bakgrunnsfarge); et vilkårlig tall større enn 0 indikerer en ikke-tom pixel (vi antar at alle slike tall tilsvarer 'svart' farge). Dersom det er en linje fra $(x1, y1)$ til $(x1, y2)$, vil alle pixlene $(x1, y1), (x1, y1 + 1), (x1, y1 + 2) \dots (x1, y2)$ være farget (ha verdi større enn 0).

I oppgaven skal vi ikke manipulere eksplisitt grafiske objekter men vi skal jobbe med en "matrise-representasjon". Denne skal leses fra en fil der matrisen er representert i tekst ved at hver ikke-tom pixel er merket med et tall (større enn 0). Filen har følgende format:

$x$ $y$
$x_1$ $y_1$ $v_1$
$x_2$ $y_2$ $v_2$
osv. ...

Den første linjen angir matrise dimensjoner $x : y$. Deretter kommer tripler av heltall: en trippel $x_i$ $y_i$ $v_i$ sier at pixel $x_i : y_i$ er merket med verdi $v_i$ (som er større enn 0 – tomme pixlene, med verdi 0, er ikke med på filen).

Du har tre filer `peter/???/???` .... som du skal bruke for å teste programmet ditt.

# 3 Sekvens-representasjon

Programmet ditt skal lese filer med matrise-representasjon av input men skal utføre beregninger på en annen, intern representasjon som vi kaller "sekvens-representasjon". Denne er som følger.

Under antakelse at alle bygningene står på nederste nivå – her 6, kan man representere en bygning ved tre tall, f.eks. tre bygningene på 'Input' tegning er hhl. $(1, \mathbf{0}, 4)$, $(2, \mathbf{3}, 6)$ og $(3, \mathbf{2}, 7)$, der første tallet angir venstre vegg, midterste tallet, $\mathbf{x}$, høyden, og siste tallet høyre vegg.

En 'SkyLine' kan representeres som en sekvens av tall, her $(1, \mathbf{0}, 4, \mathbf{2}, 7)$.

Legg merke til at denne representasjonen antar at vi har å gjøre med bygninger, dvs. tre rette linjer som danner (en del av) en rektangel. Dersom matrisen inneholder noen "løse", er ikke disse relevante for denne representasjonen.

# 4 Oppgaven

går nå ut på lage et program som

1. leser inn en matrise-representasjon (fra en fil),
2. konverterer denne til sekvens-representasjon,
3. beregner `skyLine` og
4. skriver ut resultatet.

Det er et essensielt tilleggskrav til pkt. 3 som omtales under i 4.3.

## 4.1 Datastruktur

Design datastruktur for hele modulen. Spesifiser ev. *datainvarianter*.

`s)` Datastrukturen vil måtte representere sekvenser av tall tilsv. 3 på en eller annen måte – vi kalte denne for "sekvens-representasjon". Beregning av `skyLine` vil skje på denne representasjonen.

`m)` Men det vil også være bruk for en matrise-representasjon – det vil, for eksempel, være hensiktsmessig å lese inn data fra fil til en intern matrise, og så konvertere denne til sekvens-representasjonen.

## 4.2 Konvertering

Design en algoritme som konverterer en matrise-representasjon `m)` til sekvens-representasjon `s)`. Besvarelsen til denne deloppgaven skal være en tilstrekkelig pseudo-kode for algoritmen.

## 4.3 SkyLine

Design en $\mathcal{O}(n * log(n))$ algoritme for å beregne en `skyLine`, der $n$ er antall bygninger. Denne algoritmen skal arbeide med sekvens-representasjonen `s)` tilsv. 3, dvs. input er en sekvens av $n$ byggninger, hver representert som 3 tall, og output er en sekvens av tall tilsvarende en `skyLine`. Besvarelsen til denne deloppgave skal være en tilstrekkelig pseudo-kode for algoritmen.

## 4.4 Output

Bestem hvordan du vil vise resultater på skjermen. Det er sikkert ikke så lurt å innlate seg inn på en "sikkelig" grafisk fremstilling. Programmet kan vise en liste av tall tilsvarende en `skyLine` som beskrevet i 3 eller "tegne" en matrise på skjermen ved å skrive den ut rad-for-rad med passende antall blanke tegn for hver tom pixel og et valgt tegn for de ikke-tomme.

# 5 Implementasjon

## 5.1

Implementer alle bitene fra 4 i en modul med følgende grensesnitt (du kan ev. tilpasse parameterlistene dine behov):

1. **public** $SkyLine(...)$ – konstruktør med passende initialisering;
2. **public** *void* $les(File f)$ – leser inn matrise fra filen $f$;
3. **public** ??? $skyLine()$ – beregner `skyLine` for en by lagret i datastrukturen; hva som returneres her, får du velge selv men meningen er at en bruker av modulen kan laste inn en by, beregne `skyLine`, hente/aksessere resultatet, laste inn en ny by, osv.

## 5.2 Test

Lag så et program som bruker modulen og tillater en å teste den.