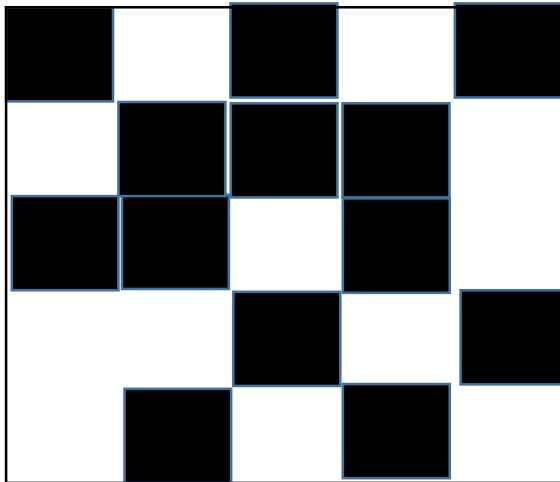


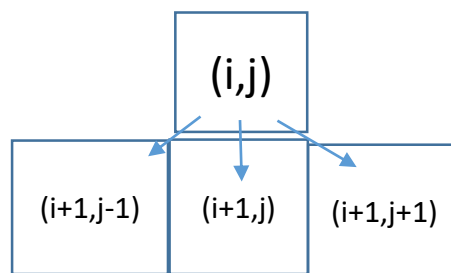
### Esercizio 1 del 16/4/2018

Dato un array `bool N[400]`, lo vediamo come una matrice  $X[m][m]$  (con  $m*m \leq 400$ ). Le caselle true di  $X$  sono posizioni aperte, mentre le false sono chiuse. Si cerca un cammino che consiste solo di caselle aperte (true) e che conduce dalla prima all'ultima riga di  $X$ . Vediamo un esempio nel quale per semplicità disegniamo bianche le caselle aperte e nere quelle chiuse.

**Esempio.** Sia  $X$  la matrice che segue. Esiste un cammino di caselle bianche che porta dalla prima riga all'ultima e che consiste delle seguenti caselle: (0,3) (1,4) (2,4) (3,3) (4,2)



Come si osserva da questo esempio, il cammino deve andare sempre avanti, cioè deve avere lunghezza pari al numero delle colonne. Inoltre, se siamo nella casella  $(i,j)$ , la prossima casella bianca può essere solo  $(i+1,j-1)$  o  $(i+1,j)$  oppure  $(i+1,j+1)$ . Insomma la da  $(i,j)$  il cammino può proseguire solo nei modi seguenti:



Si chiede di scrivere 2 funzioni ricorsive che cercano un tale cammino e restituiscono true sse il cammino c'è. La prima funzione è partenza e deve soddisfare la seguente specifica:

`PRE_p=(N ha  $m*m$  valori definiti,  $0 \leq i \leq m$ )`

`bool partenza(bool*N, int m, int i)`

`POST_p=(risponde true sse esiste un cammino dalla prima all'ultima riga di  $N$  vista come  $X[m][m]$  e che inizia in un elemento tra  $i$  e  $m-1$  della prima riga di  $X$ )`

La funzione partenza, una volta fissato il punto di partenza della prima riga ( $i$ ), invoca un'altra funzione trova sempre ricorsiva che soddisfa la seguente specifica:

PRE\_t=(N ha  $m*m$  elementi definiti)

bool trova(bool\* N, int m, int i, int j)

POST=(restituisce true se in N vista come  $X[m][m]$  c'è un cammino da (i,j) alla riga m-1 composto da sole caselle bianche)

**Correttezza:**

- 1) Dimostrare induttivamente la correttezza di trova rispetto a PRE\_t e POST\_t.
- 2) Dimostrare induttivamente la correttezza di partenza rispetto a PRE-p e POST\_p.