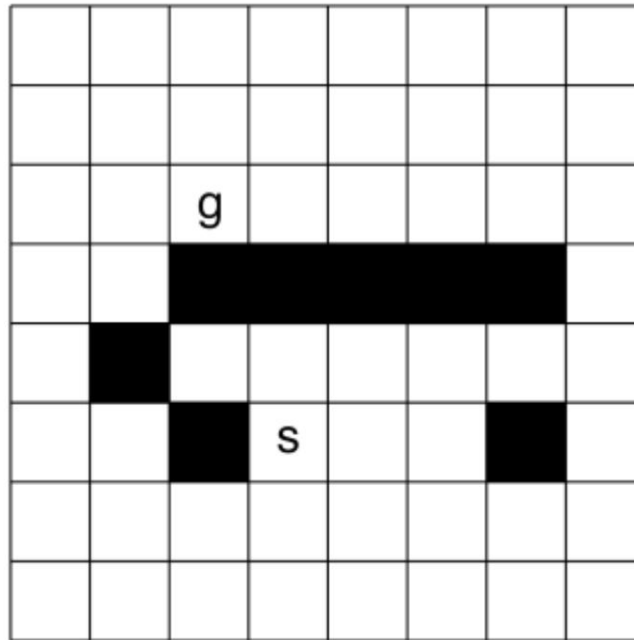


Compito 2: Risoluzione dei problemi come ricerca

Consideriamo il problema di trovare un percorso nella griglia mostrata nella figura seguente dalla posizione s alla posizione g.

Un pezzo può muoversi sulla griglia orizzontalmente o verticalmente, una casella alla volta. Non è consentito entrare nelle zone nere vietate.



Scegli uno dei seguenti casi da risolvere e scrivi un programma Python per implementarlo, mostrando come vengono esplorati i nodi e come vengono espansi i percorsi:

1. Sulla griglia mostrata sopra, numerare i nodi espansi (in ordine) per una ricerca approfondita da s a g, dato che l'ordine degli operatori è su, sinistra, destra e giù. Supponiamo che ci sia potatura del ciclo. Qual è il primo percorso trovato?
2. Su una copia della stessa griglia, numerare i nodi espansi, in ordine, per una golosa ricerca best-first da s a g. Come funzione di valutazione dovrebbe essere utilizzata la distanza di Manhattan. La distanza di Manhattan tra due punti è la distanza nella direzione x più la distanza dentro la direzione y. Corrisponde alla distanza percorsa lungo le strade cittadine disposte a griglia. Assumere l'eliminazione di percorsi multipli. Qual è il primo percorso trovato?
3. Su una copia della stessa griglia, numerare i nodi espansi, in ordine, per una ricerca euristica in profondità da s a g, data la distanza di Manhattan come funzione di valutazione. Assumere la potatura del ciclo. Qual è il percorso trovato?
4. Numerare i nodi in ordine per una ricerca A*, con potatura di percorsi multipli, per la stessa griglia. Qual è il percorso trovato?

Nota: 1- È necessario tracciare la soluzione animata, ovvero mostrare il movimento dell'agente passando da una cella all'altra fino ad arrivare alla cella obiettivo.
 2- È necessario tracciare il grafico che rappresenta lo spazio del problema, dove i nodi sono le celle e gli archi sono le azioni (move_up, move_down, move_left, move_right), che potrebbero essere associate ai costi.