

AI LAB 2022-23

Progetti Prolog e ASP

Frigato Luca

Strategie di Ricerca informata su Labirinto

- **A***
 - Implementazione a partire dalle regole di **BFS**
- **IDA* (Iterative Deepening A*)**
 - Implementazione a partire dalle regole di **IDFS**

Esperimenti

- Variazione dimensioni massima, euristica, densità muri. Grazie a TOOL creato per creare, visualizzare le mappe (e le soluzioni a vari step temporali calcolate)
- Gli esperimenti si sono svolti su casi **di TEST con GOAL multipli**

Provate 3 Euristiche Euclidea - Manhattan – Chebyshev

```
/ progetto / src / euristiche / Euclidea.pl
%Distanza Euclidea: Calcola la distanza Euclidea tra lo stato
le coordinate.

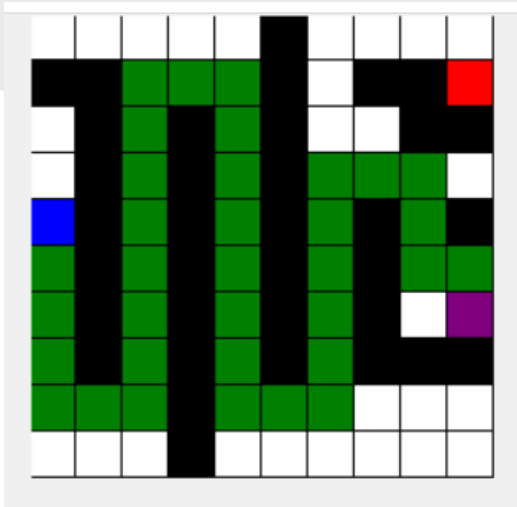
heuristic(pos(R1, C1), pos(R2, C2), Distance) :-
    Distance is (R1 - R2)^2 + (C1 - C2)^2,.
```

```
og > progetto > src > euristiche > Manhattan.pl
% Calculate the Manhattan distance between two positions
heuristic(pos(R1, C1), pos(R2, C2), Distance) :-
    Distance is abs(R1 - R2) + abs(C1 - C2).
```

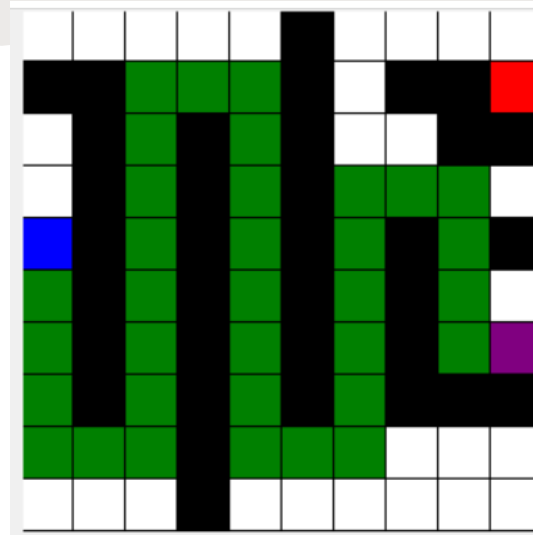
```
% Calculate the heuristic H for the current state against all possible final states
calculate_heuristics(pos(Riga, Col), MinimumHeuristic) :-
    findall(Heuristic, (
        finale(pos(RFin, ColFin)),
        heuristic(pos(Riga, Col), pos(RFin, ColFin), Heuristic)
    ), Heuristics),
    min_list(Heuristics, MinimumHeuristic).
%write('Calcolo euristica per pos(', write(Heuristics), nl.
```

Esperimenti:

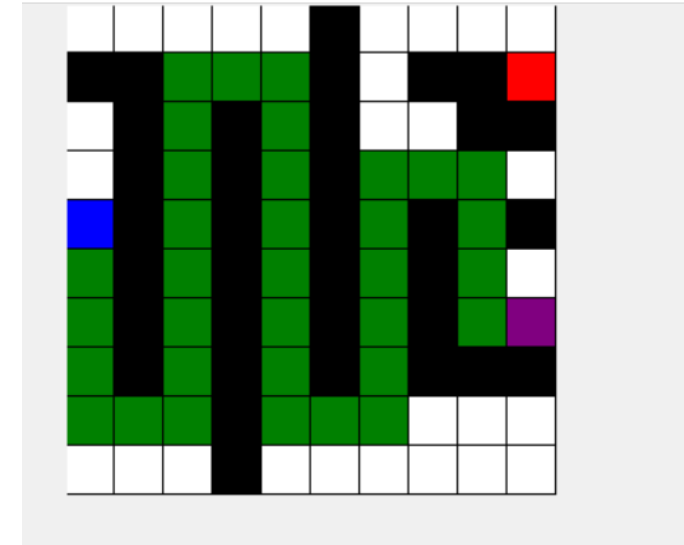
SPECS: CPU i7 7500u (2 core 2 thread)
RAM: 16 GB (2 riservati)



A*



IDFS*

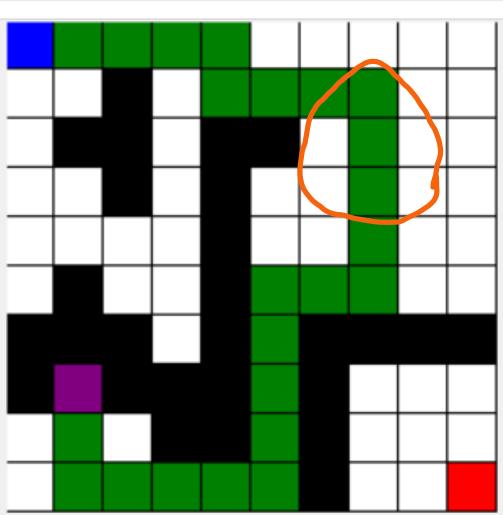


IDA*

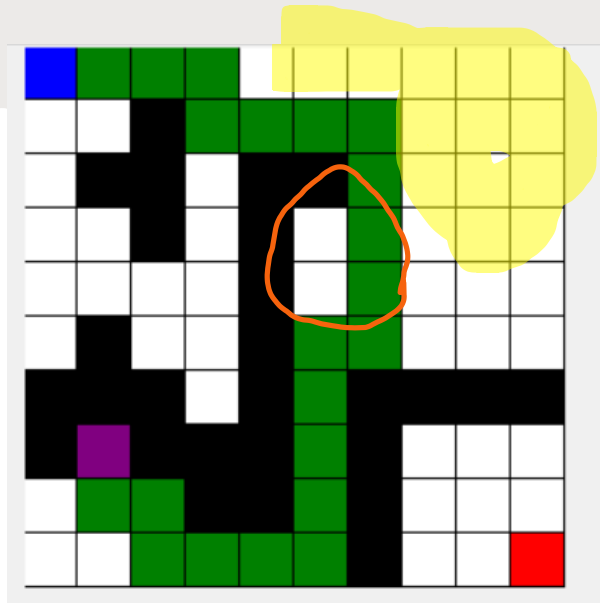
Facile: la mappa favorisce la ricerca in profondità (i muri aiutano a limitare la ricerca sull'unico path disponibile).

	A*	IDFS	IDA*
Time	Istantaneo	istantaneo	istantaneo
Steps	35	35	35

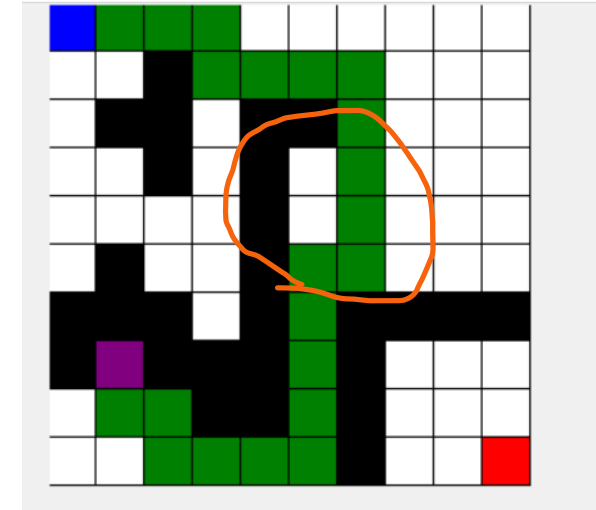
Esperimenti:



A*



IDFS*

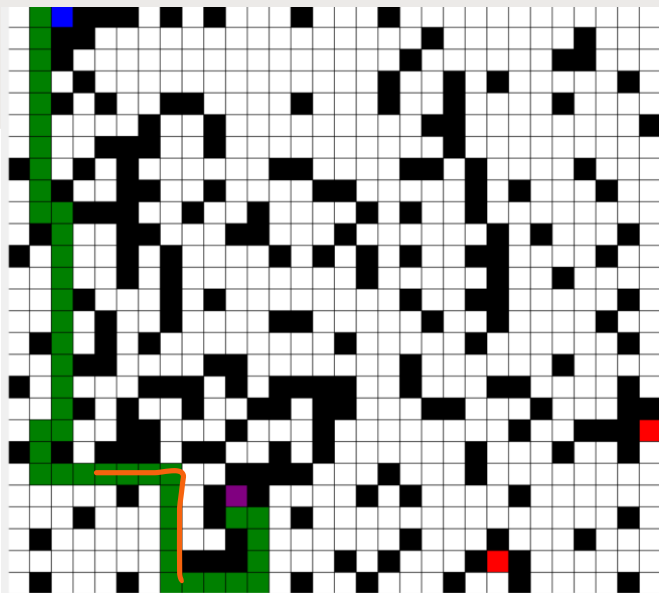


IDA*

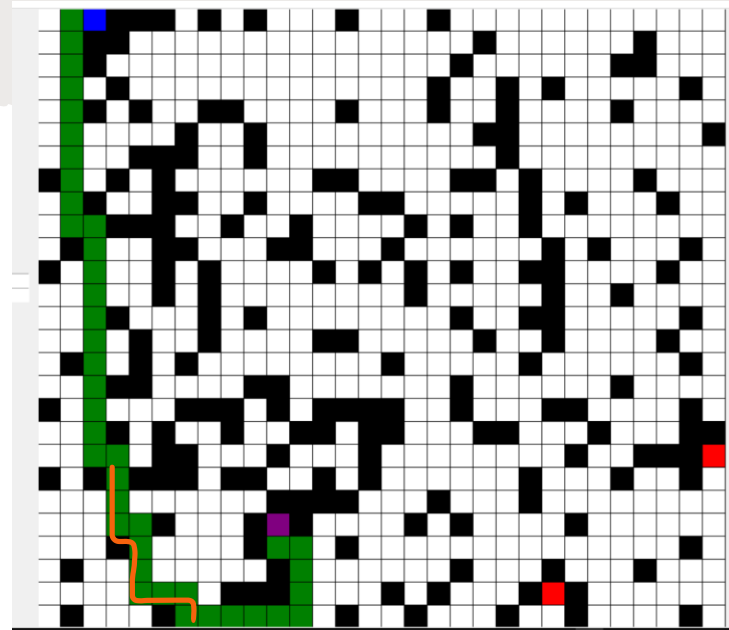
Fattore di incertezza introdotto dal fatto che ora il branching factor (il numero di nodi figli di ogni nodo) è aumentato considerevolmente e con questo la profondità massima

	A*	IDFS	IDA*
Time	Istantaneo	<30 secondi	istantaneo
Steps	36	35	35

Esperimenti:



A*



IDA*

Fattore sparsità diventa molto importante per IDFS che si perde nella ricorsione.

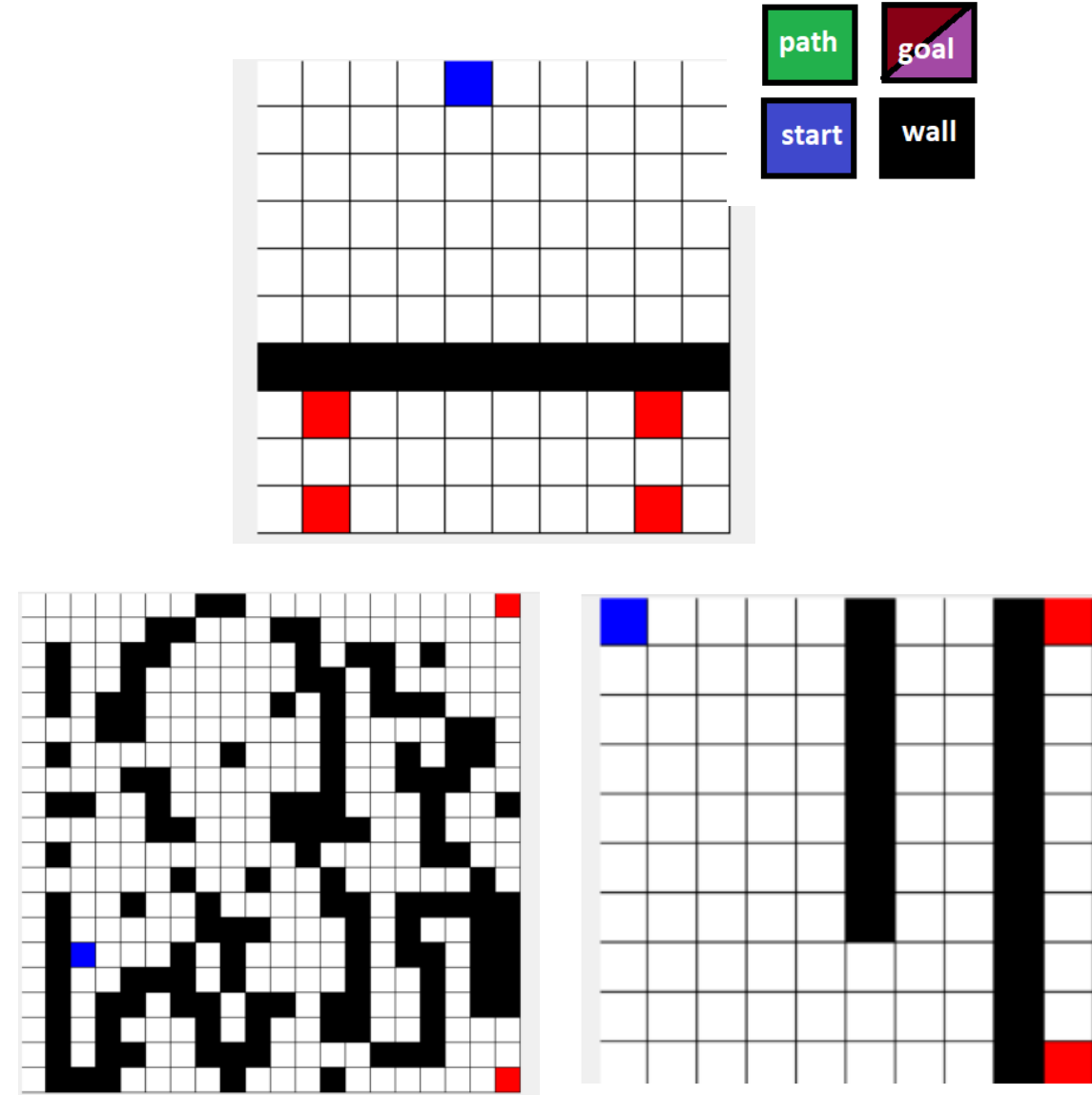
IDA nonostante impieghi più tempo di A*, trova una soluzione migliore ma con mooolte esplorazioni

```
51449 New iteration...
51450 [pos(9,3),pos(10,3),pos(10,4),pos(10,5),pos(10,6),pos(9,6),pos(8,6),pos(7,6),pos(6,6),pos(5,6),pos(4,6),pos(3,6),pos(2,6),pos(2,5),pos(2,4),pos(1,4),pos(1,3),pos(1,2),pos(1,1)]
```

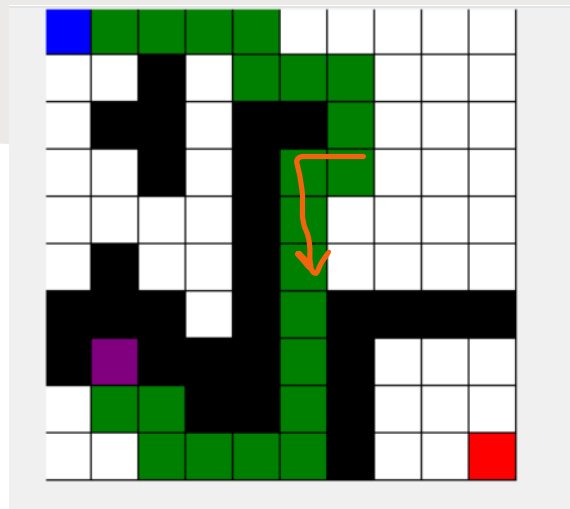
	A*	IDFS	IDA*
Time	Istantaneo	Stoppato dopo 30 minuti	3 minuti
Steps	44	35	42

Esperimenti: no soluz

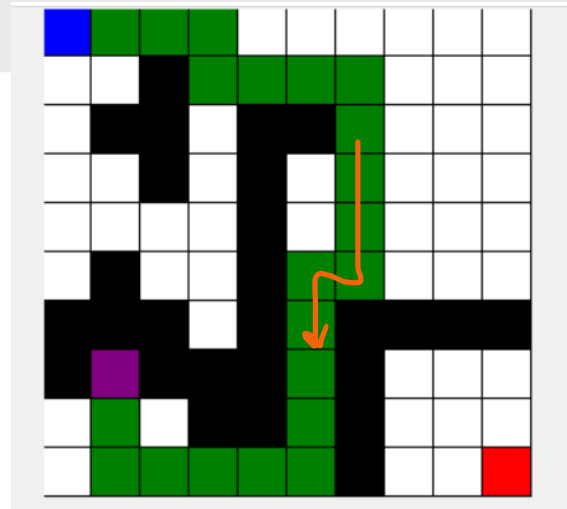
- **Risultato teorico:**
- L'algoritmo IDDFS non termina in presenza di labirinti senza soluzione (Figura 1f). Non avendo impostato una soglia massima infatti, l'algoritmo procede incrementando infinite volte la profondità massima a cui cercare una soluzione. Al contrario, gli algoritmi IDA* e A* forniscono la garanzia di terminazione se lo spazio di ricerca è finito, ritornando false in assenza di soluzione. Ci impiegano però comunque molto tempo.



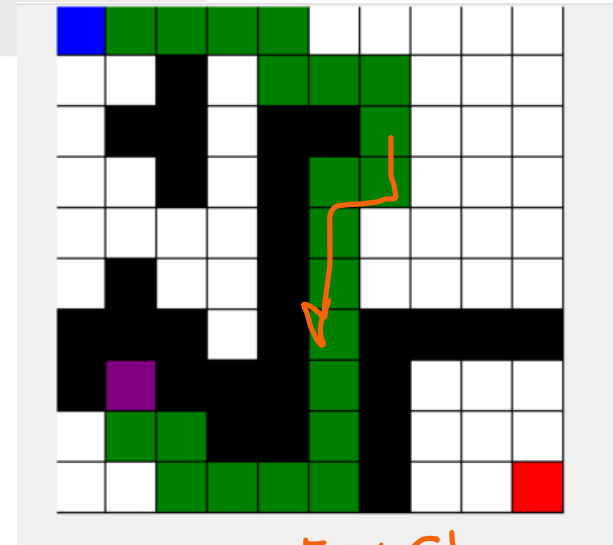
Euristiche



M



CH



EUC

L'euristica che portato a risultati più consistenti è la distanza di **Manhattan** mentre con quella di Chebyshev o Euclidea i tempi si dilatano notevolmente con l'aumentare della difficoltà della mappa. La cause dovrebbero essere le operazioni floating point o di max per il calcolo della diff..

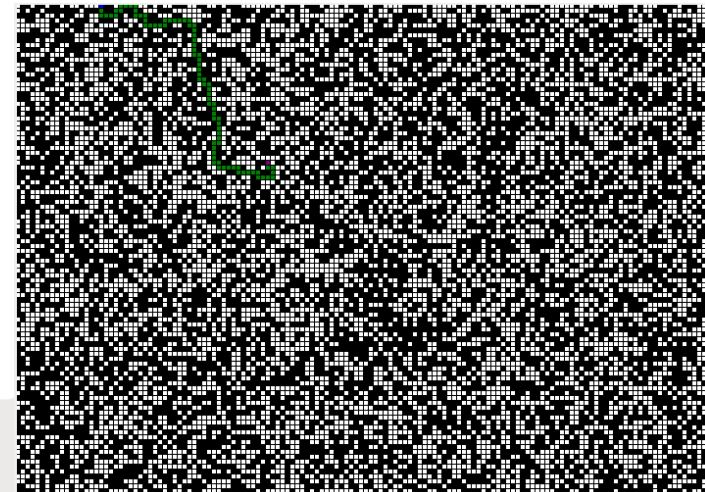
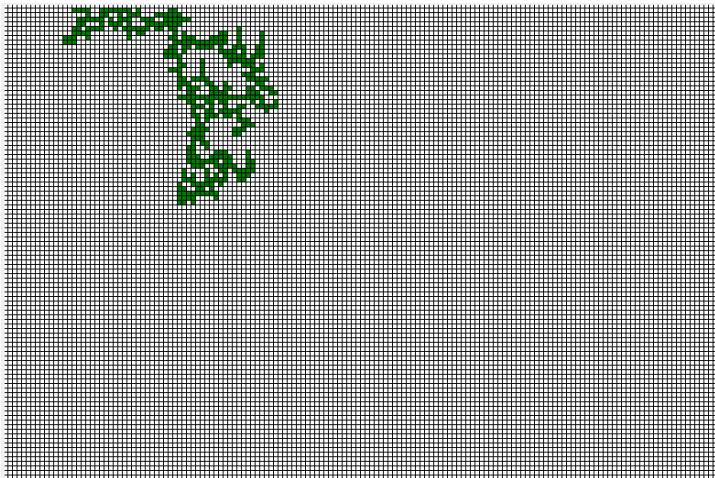
Considerazioni finali



- Tempisticamente, A^* si comporta meglio nonostante il costo in memoria
- Le soluzioni ottenute sono invece comparabili e molto simili
- IDA* riparte sempre dalla radice prima di riscendere per effettuare un altro taglio, e questo comporta latenze con l'aumento della complessità del labirinto
- Gli algo restituiscono il path verso lo stato goal piu vicino allo stato di partenza.

A^* si è comportato anche molto bene nel 160x160

IDA* verificando con la trace, si è perso nel considerare l'area a sx della posizione iniziale.

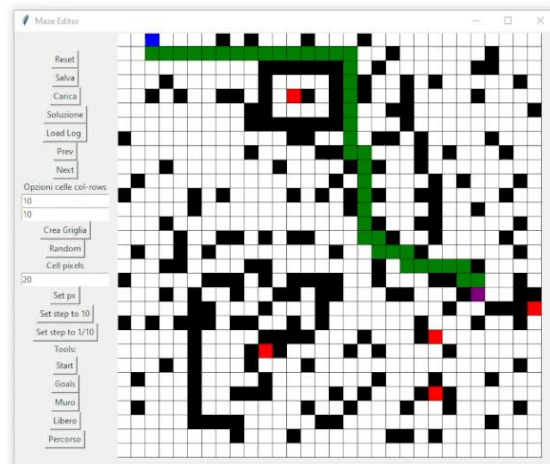
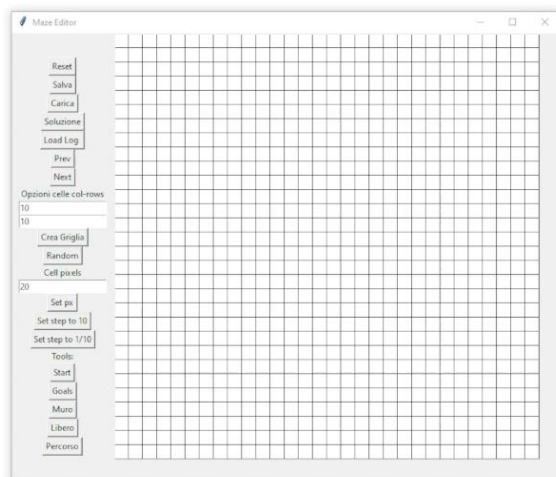


Risultati finali

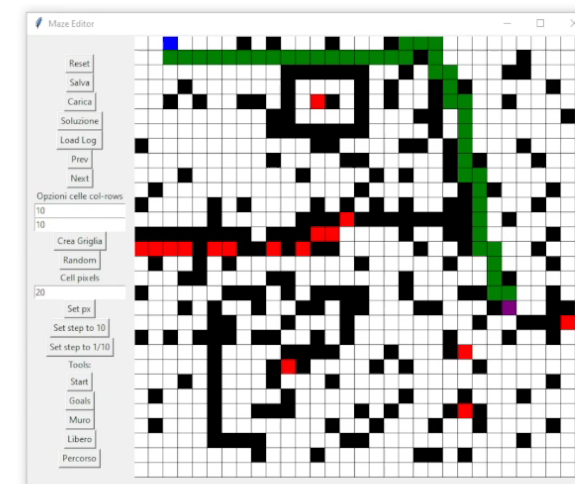
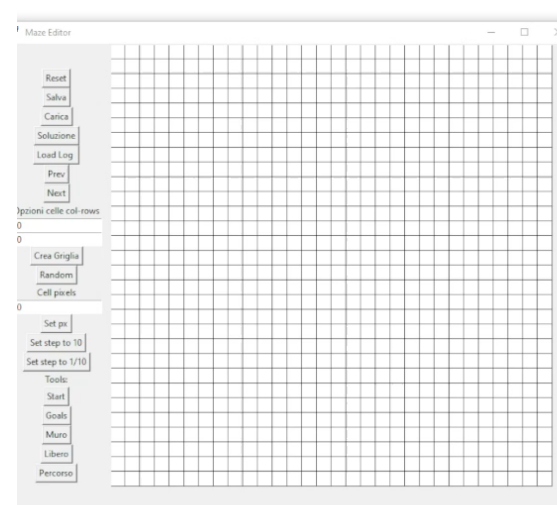
A* in steps
- multi GOAL



ok,num_fire_ko,num_guess_ok,num_guess_ko,num_safe,num_sink
,7,2,8)
,1,9)



ok,num_fire_ko,num_guess_ok,num_guess_ko,num_safe,num_sink
,7,2,8)
,1,9)

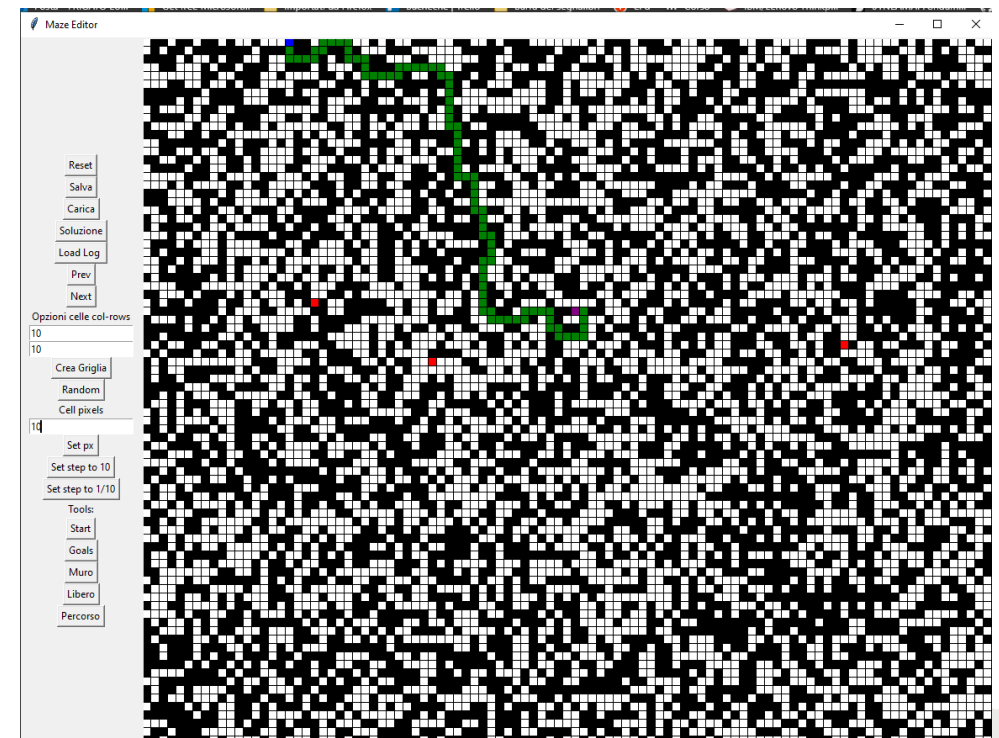
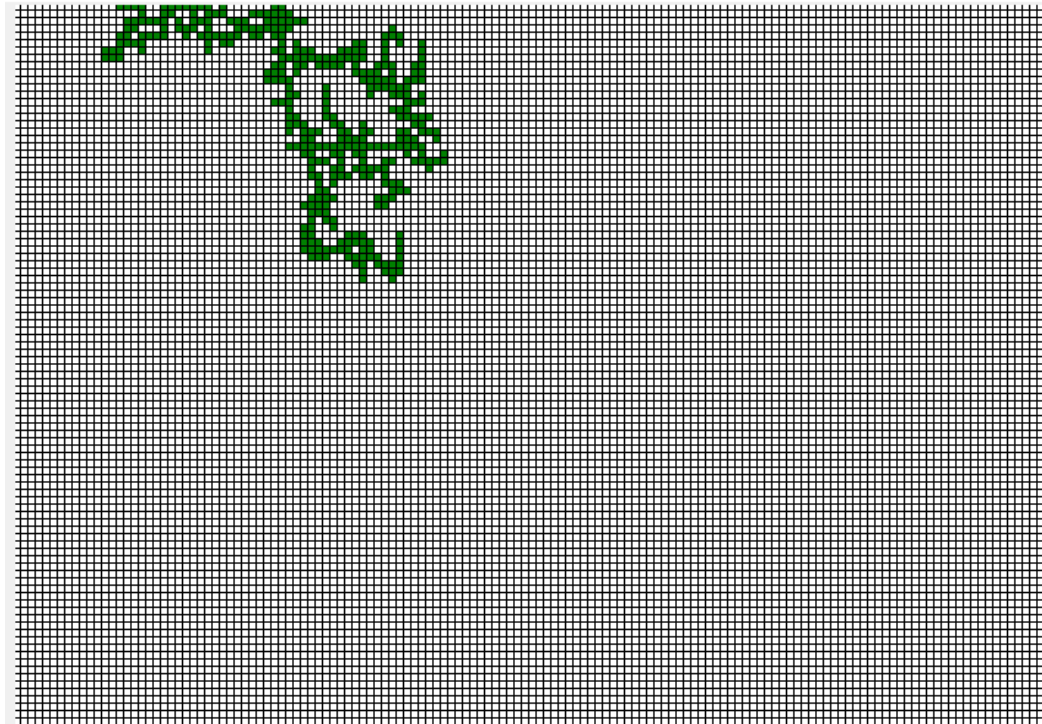


Considerazioni finali



A* in steps

- **160x160** risolto in una decina di secondi
 - **1000x1000 heap dump**...finita la ram
- > **limiti di A***



ASP

Calendario sportivo

Come in mondiali.cl, si è reso necessario pre assegnare le squadre alle città ed utilizzare degli aggregati per imporre in vincoli (4)-(7)

```
% 1. Definizione delle squadre  
squadra(a;b;c;d;e;f;g;h;i;j;k;l;m;n;o;p;q;r;s;t).  
  
% 2. Definizione delle giornate  
giornata(1..38).  
  
% 3. Definizione delle città e mapping sulla squadra  
citta(usfq; zviz; wisw; uwiw; azwk; vrge; dfjd; irzi; zkpc; ngma; ndgg; rtsw; xuwr; xuzx; cmgb; vinm; nextb; fkhc; uhmg; ).
```

ASP

Calendario sportivo

```
afferisce(a,usfq).           % Due delle 20 squadre fanno riferimento alla medesima città
afferisce(t,usfq).
afferisce(b,zviz).
afferisce(s,wisw).
afferisce(c,vrge).
afferisce(r,uwiw).
afferisce(d,dfjd).
afferisce(q,azwk).
afferisce(e,irzi).
afferisce(p,zkpc).
afferisce(f,ngma).
afferisce(o,ndgg).
afferisce(g,rtsw).
afferisce(n,xuwr).
afferisce(h,xuzx).
afferisce(m,cmgb).
afferisce(i,vinm).
afferisce(l,nxtb).
afferisce(j,fkhc).
afferisce(k,uimg).
```

ASP

Calendario sportivo

```
% Ogni squadra gioca MAX due volte contro tutte le altre squadre (una in casa e una fuori)
1 {gioca(S,A,C,N):citta(C),giornata(N)} 1 :- squadra(S),squadra(A),S<>A,afferisce(S,C).
1 {gioca(A,S,C,N):citta(C),giornata(N)} 1 :- squadra(S),squadra(A),S<>A,afferisce(A,C).
```

```
% Violazione di vincolo se squadre diverse della stessa citta possono giocare in casa nella stessa giornata.
% Quindi violazione se S1 e S2 sono della stessa città e giocano in casa nella stessa giornata ma contro una squadra diversa.
:- squadra(S1;S2;S3),giornata(N),afferisce(S1,C),afferisce(S2,C),S1<>S2,gioca(S1,S3,C,N),gioca(S3,S1,C,N).
:- squadra(S1;S2;S3),giornata(N),afferisce(S1,C),afferisce(S2,C),S1<>S2,gioca(S2,S3,C,N),gioca(S3,S2,C,N).
% Scritta due volte per gestire sia per S1 sia per S2
```

```
% Ciascuna squadra non deve giocare mai più di due partite consecutive in casa o fuori casa
:- squadra(S), giornata(G), incasa(S, G, PiC), fuoricasa(S, G, Pfc), N = PiC + Pfc, N > 2.

% la distanza tra una coppia di gare di andata e ritorno è di almeno 10 giornate,
:- giornata(G1), giornata(G2), G1 < G2, gioca(S1, S2, _, G1), gioca(S2, S1, _, G2), G2 - G1 < 10.
```

ASP

Calendario sportivo

SATISFIABLE

```
Models      : 1+
Calls       : 1
Time        : 3.073s (Solving: 1.56s 1st Model: 1.53s Unsat: 0.00s)
CPU Time    : 3.047s
```

[illegible]

ASP

Calendario sportivo

Matches by day (tot):

Day 1: 19 matches
Day 2: 18 matches
Day 3: 18 matches
Day 4: 17 matches
Day 5: 15 matches
Day 6: 14 matches
Day 7: 14 matches
Day 8: 11 matches
Day 9: 7 matches
Day 10: 3 matches
Day 11: 5 matches
Day 12: 9 matches
Day 13: 5 matches
Day 14: 8 matches
Day 15: 7 matches
Day 16: 3 matches
Day 17: 7 matches
Day 18: 3 matches

Day 19: 4 matches
Day 20: 3 matches
Day 21: 5 matches
Day 22: 8 matches
Day 23: 3 matches
Day 24: 7 matches
Day 25: 3 matches
Day 26: 6 matches
Day 27: 6 matches
Day 28: 5 matches
Day 29: 5 matches
Day 30: 11 matches
Day 31: 12 matches
Day 32: 16 matches
Day 33: 14 matches
Day 34: 17 matches
Day 35: 17 matches
Day 36: 18 matches
Day 37: 18 matches
Day 38: 19 matches

Total matches: 380

Matches by city:

usfq: 38 matches
zviz: 19 matches
wisw: 19 matches
vrge: 19 matches
uwiw: 19 matches
dfjd: 19 matches
azwk: 19 matches
irzi: 19 matches
zkpc: 19 matches
ngma: 19 matches
ndgg: 19 matches
rtsw: 19 matches
kuwr: 19 matches
kuzx: 19 matches
cmgb: 19 matches
vinm: 19 matches
hxtb: 19 matches
fkhc: 19 matches
uhmg: 19 matches

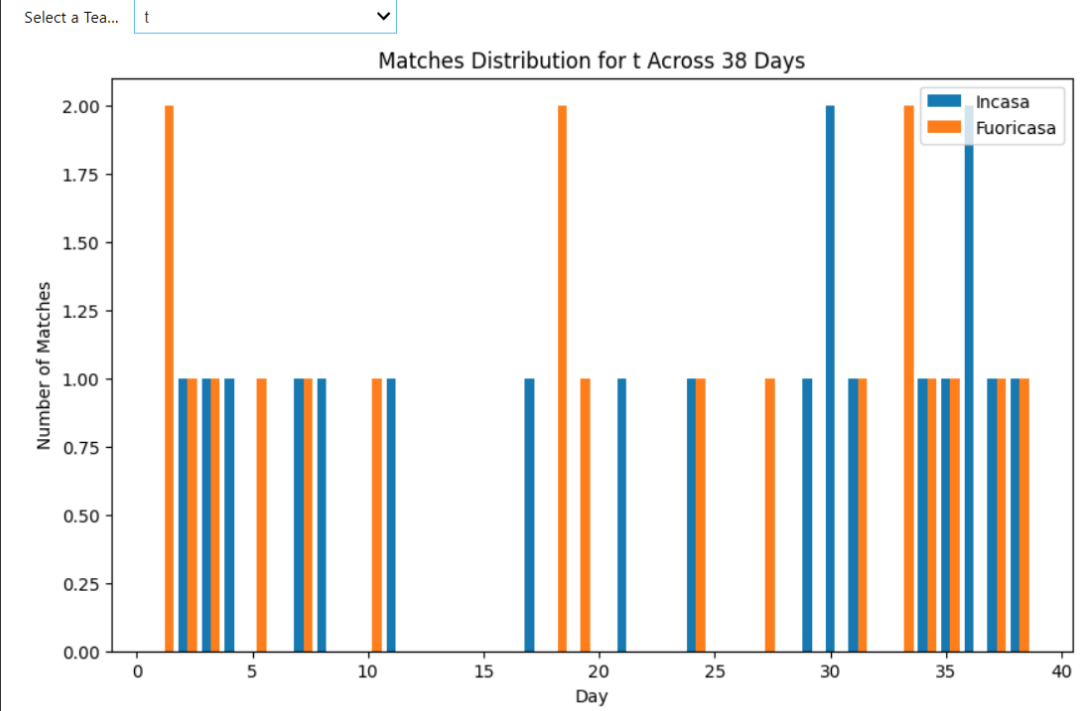
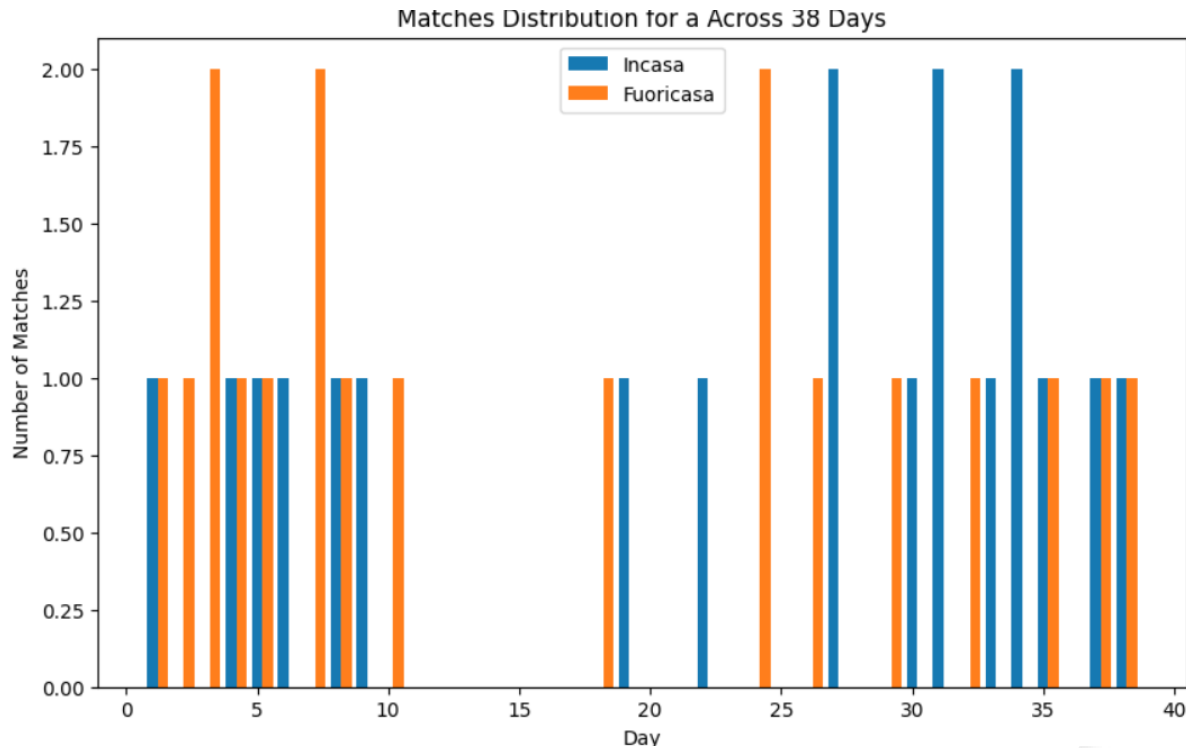
Matches by team:

a: 38 matches
b: 38 matches
c: 38 matches
d: 38 matches
e: 38 matches
f: 38 matches
g: 38 matches
h: 38 matches
i: 38 matches
j: 38 matches
k: 38 matches
l: 38 matches
m: 38 matches
n: 38 matches
o: 38 matches
p: 38 matches
q: 38 matches
r: 38 matches
s: 38 matches
t: 38 matches

DISTRIBUTION FOR A ACRC

ASP
Calendario sportivo

Incasa
Fuoricasa



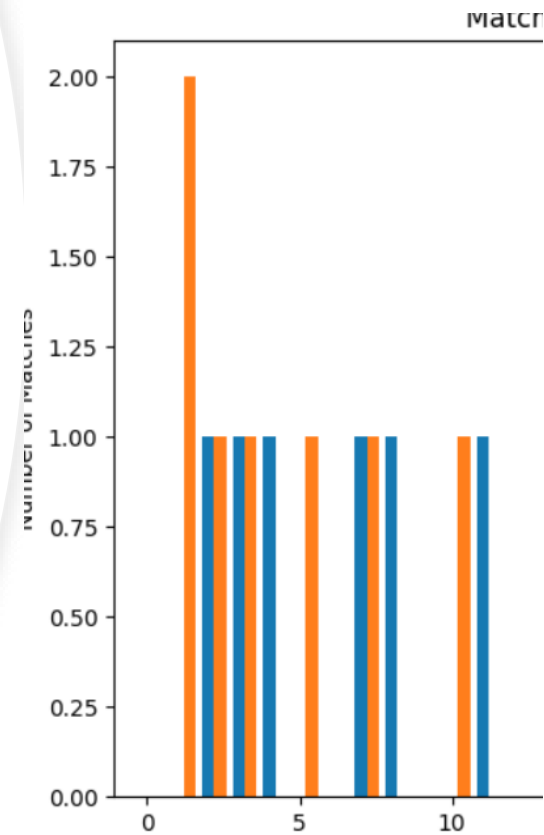
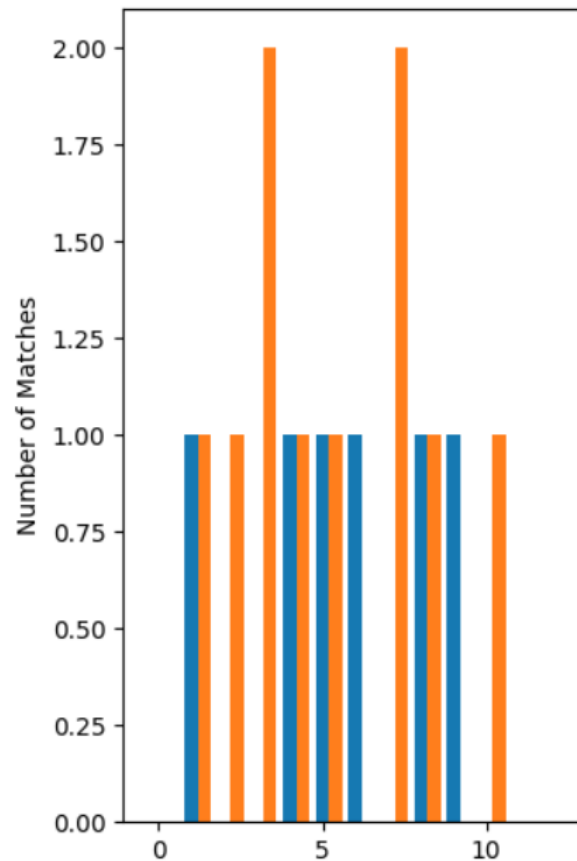
Differenza tra la distribuzione dei MATCH per A(SX) per T(DX), le due squadre della stessa città

DISTRIBUTION FOR a ACRC

ASP
Calendario sportivo

Incasa
Fuoricasa

a



Si può notare in questa sezione, che il vincolo ha funzionato. Difatti, notiamo che