

Intelligenza Artificiale I

Esercitazione 4 Ricerca nello spazio degli stati

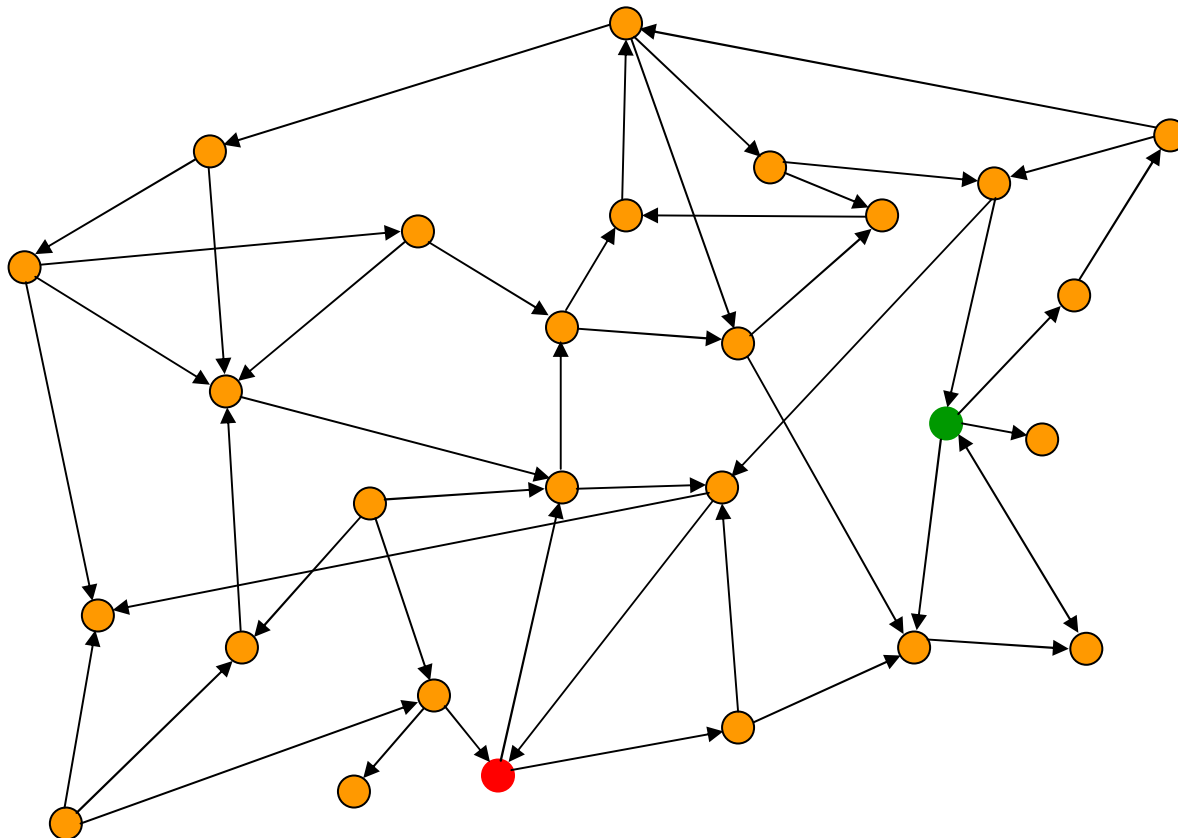
Marco Piastra

Ricerca nello spazio degli stati

- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione



Ricerca nello spazio degli stati

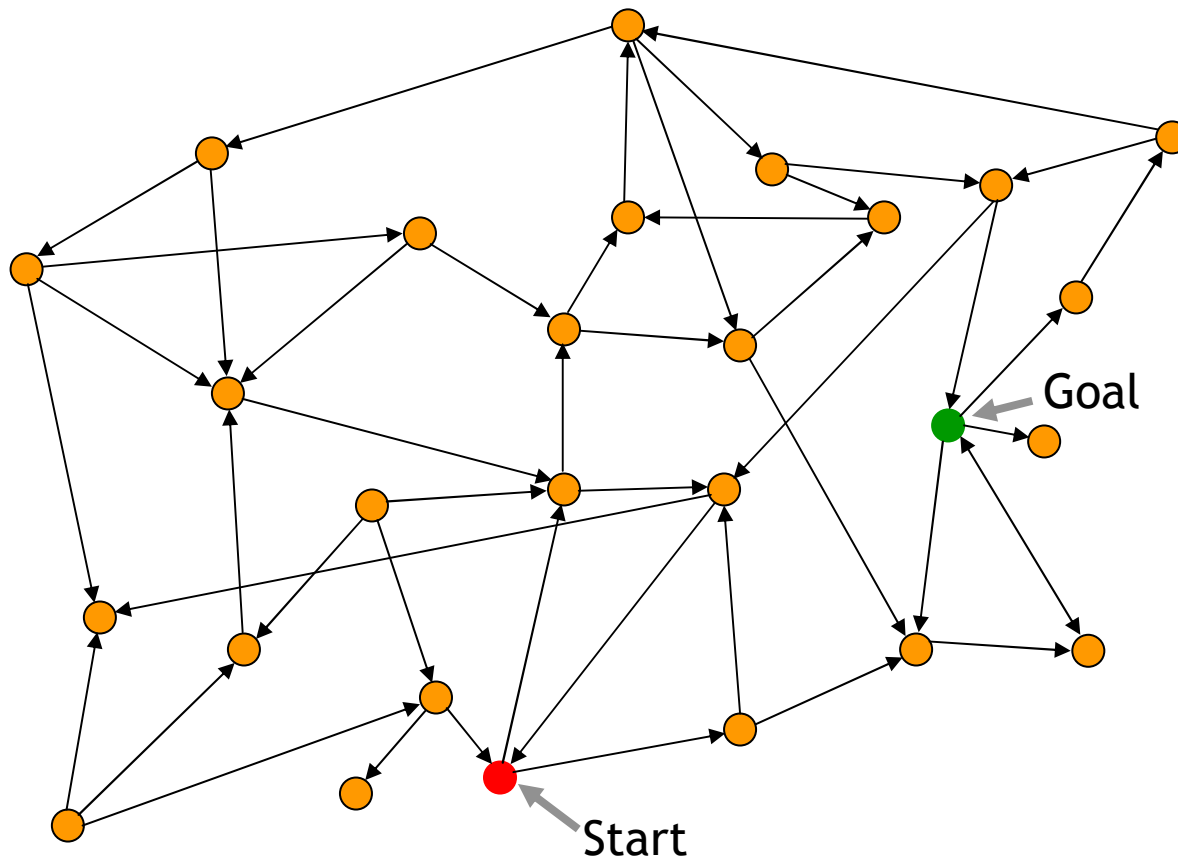
- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

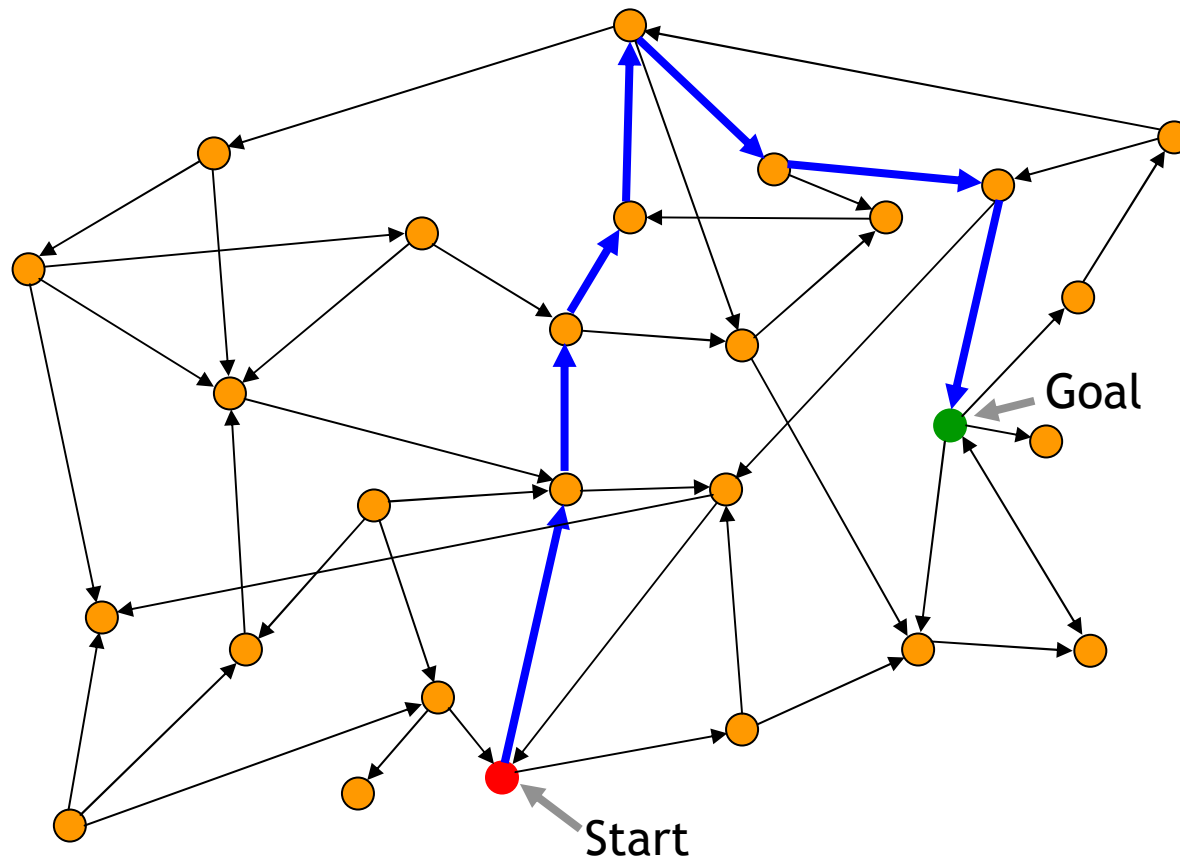
Gli archi (orientati) rappresentano una transizione

Una posizione iniziale (Start)

Una posizione finale (Goal)



Ricerca nello spazio degli stati



- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione

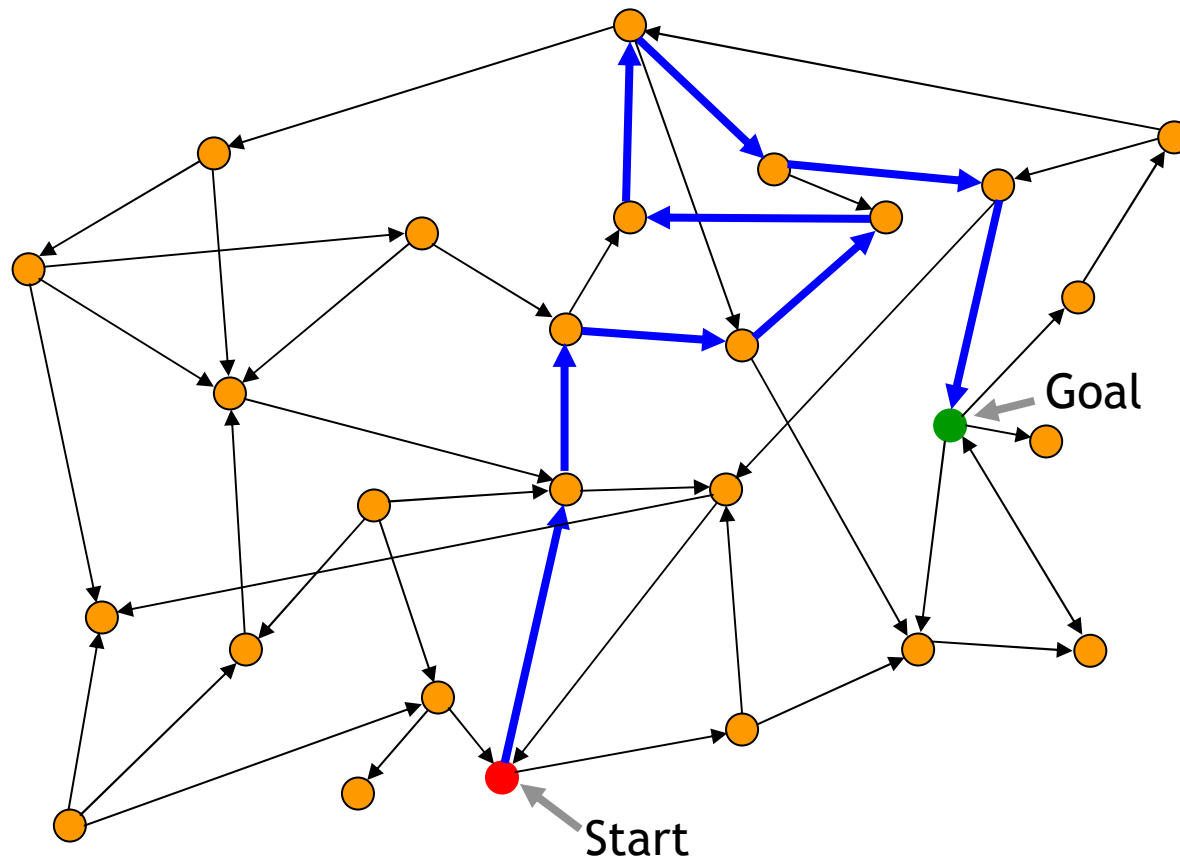
Una posizione iniziale (Start)

Una posizione finale (Goal)

Si cercano le soluzioni:
i percorsi dal nodo Start a al nodo Goal

Può bastare una sola

Ricerca nello spazio degli stati



- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione

Una posizione iniziale (Start)

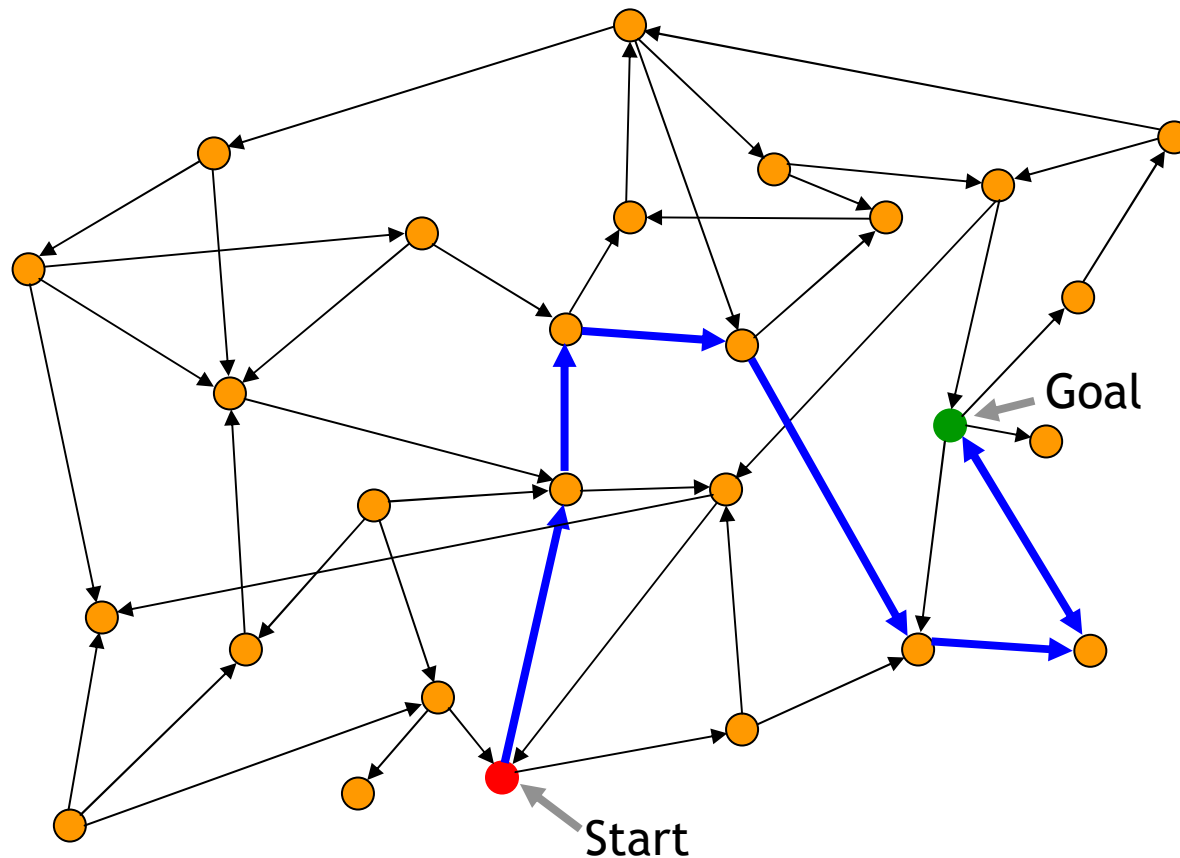
Una posizione finale (Goal)

Si cercano le soluzioni:
i percorsi dal nodo Start a al nodo Goal

Può bastare una sola

Ma può essere necessario trovarle tutte

Ricerca nello spazio degli stati



- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione

Una posizione iniziale (Start)

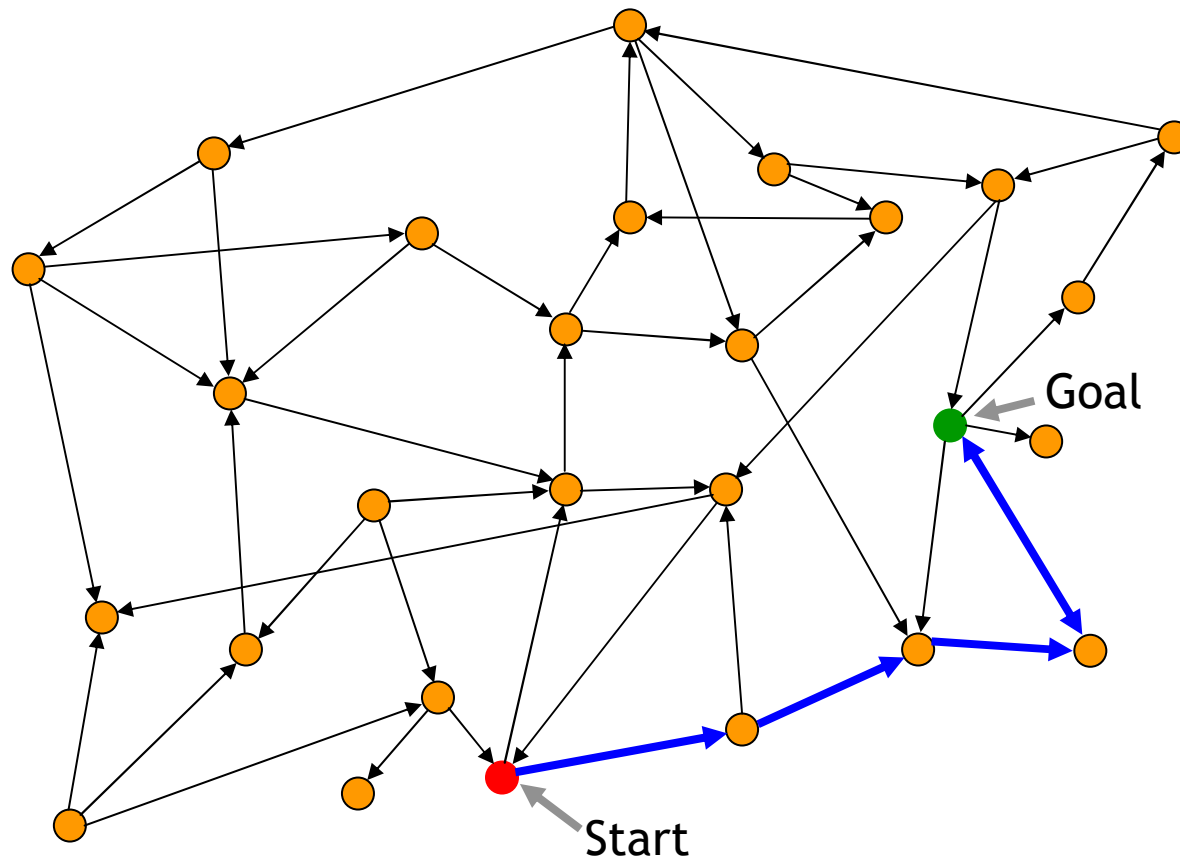
Una posizione finale (Goal)

Si cercano le soluzioni:
i percorsi dal nodo Start a al nodo Goal

Può bastare una sola

Ma può essere necessario trovarle tutte

Ricerca nello spazio degli stati



- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione

Una posizione iniziale (Start)

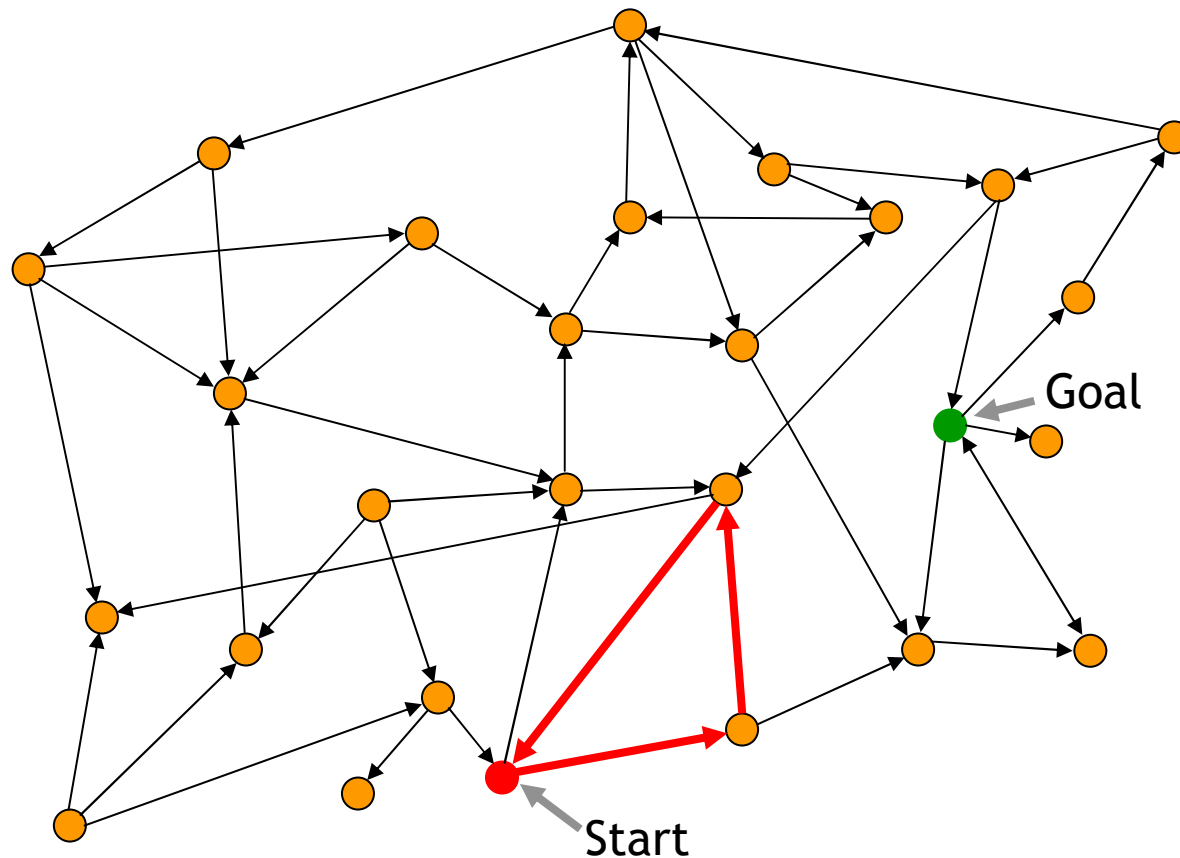
Una posizione finale (Goal)

Si cercano le soluzioni:
i percorsi dal nodo Start a al nodo Goal

Può bastare una sola

Ma può essere necessario trovarle tutte

Ricerca nello spazio degli stati



- Un grafo descrive le possibili mosse

I nodi rappresentano uno stato

Gli archi (orientati) rappresentano una transizione

Una posizione iniziale (Start)

Una posizione finale (Goal)

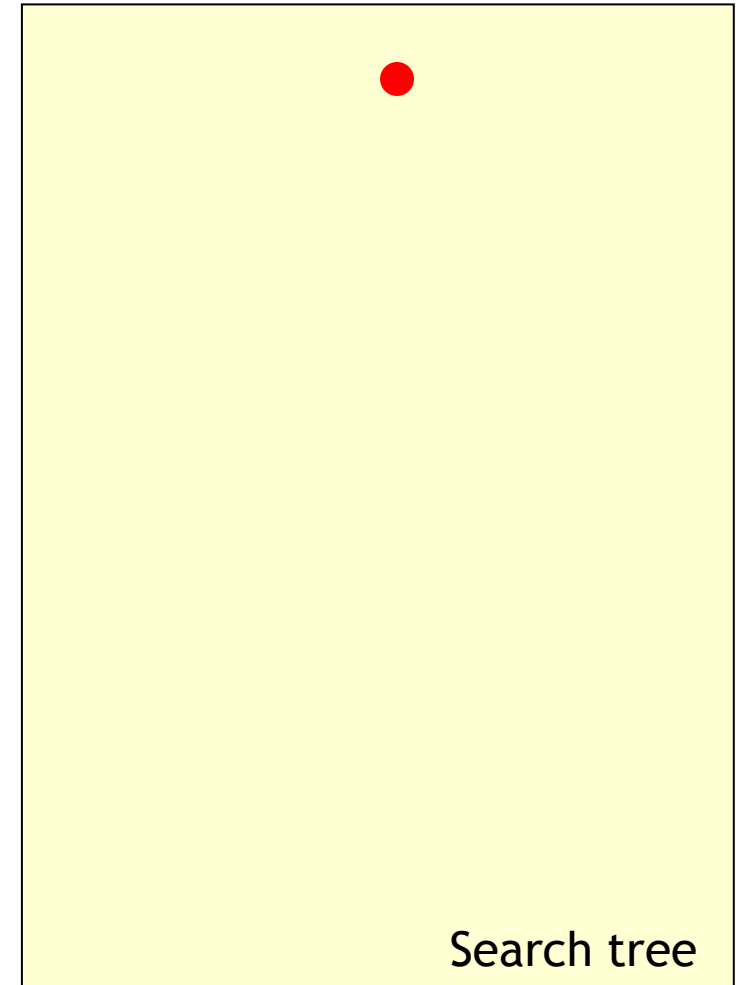
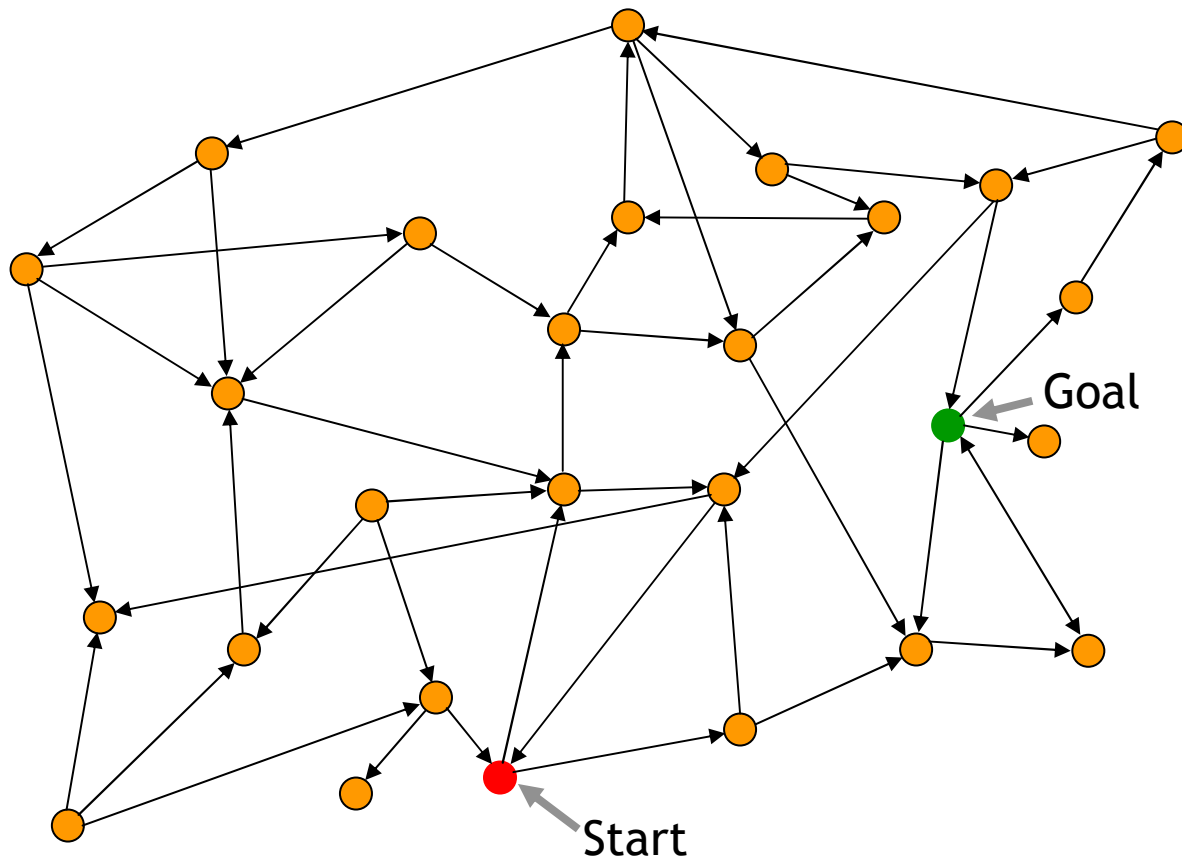
Si cercano le soluzioni:
i percorsi dal nodo Start a al nodo Goal

Può bastare una sola

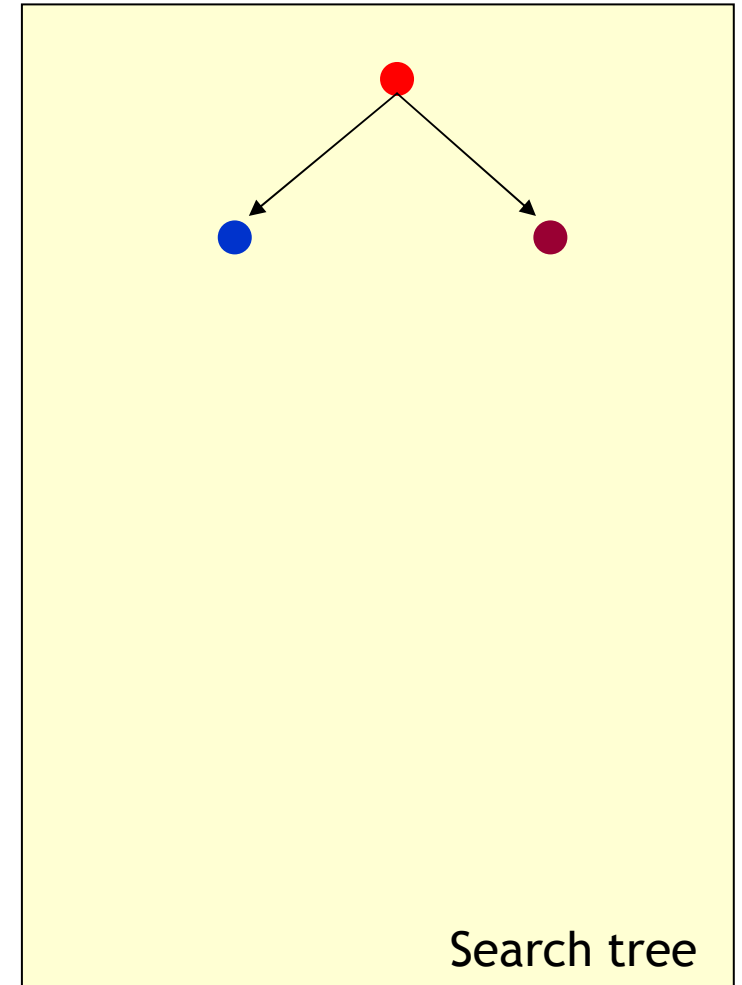
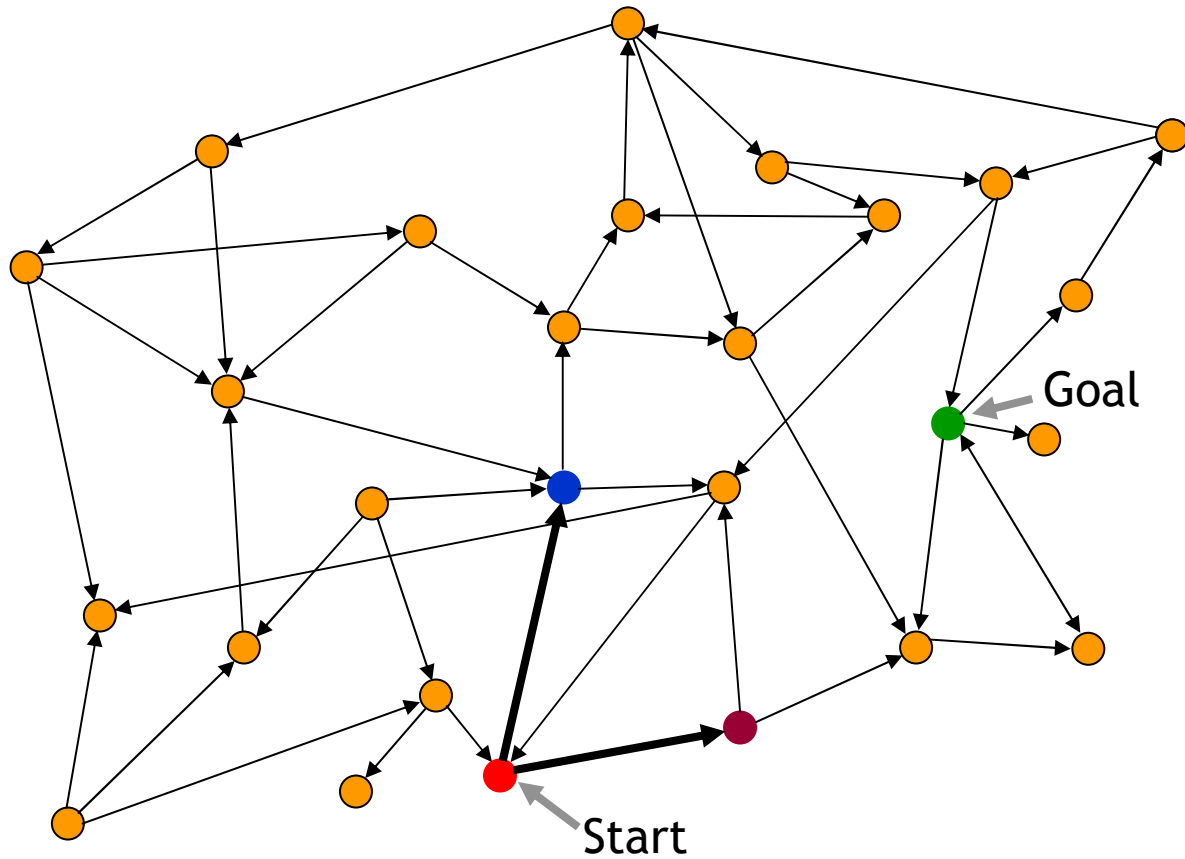
Ma può essere necessario trovarle tutte

Attenzione ai cicli!

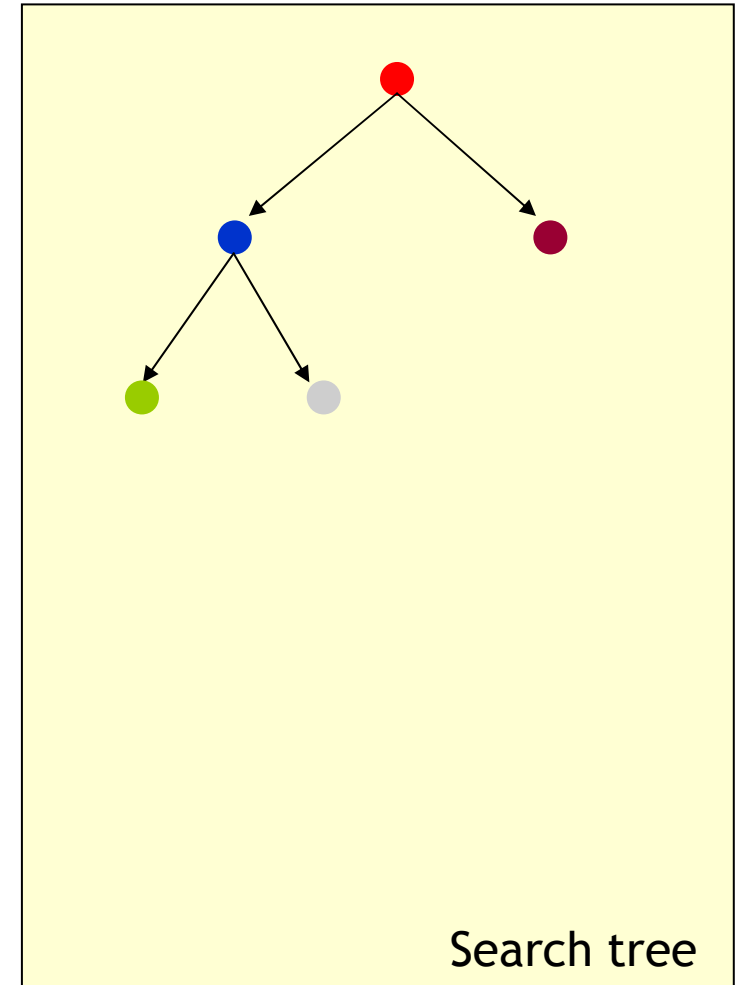
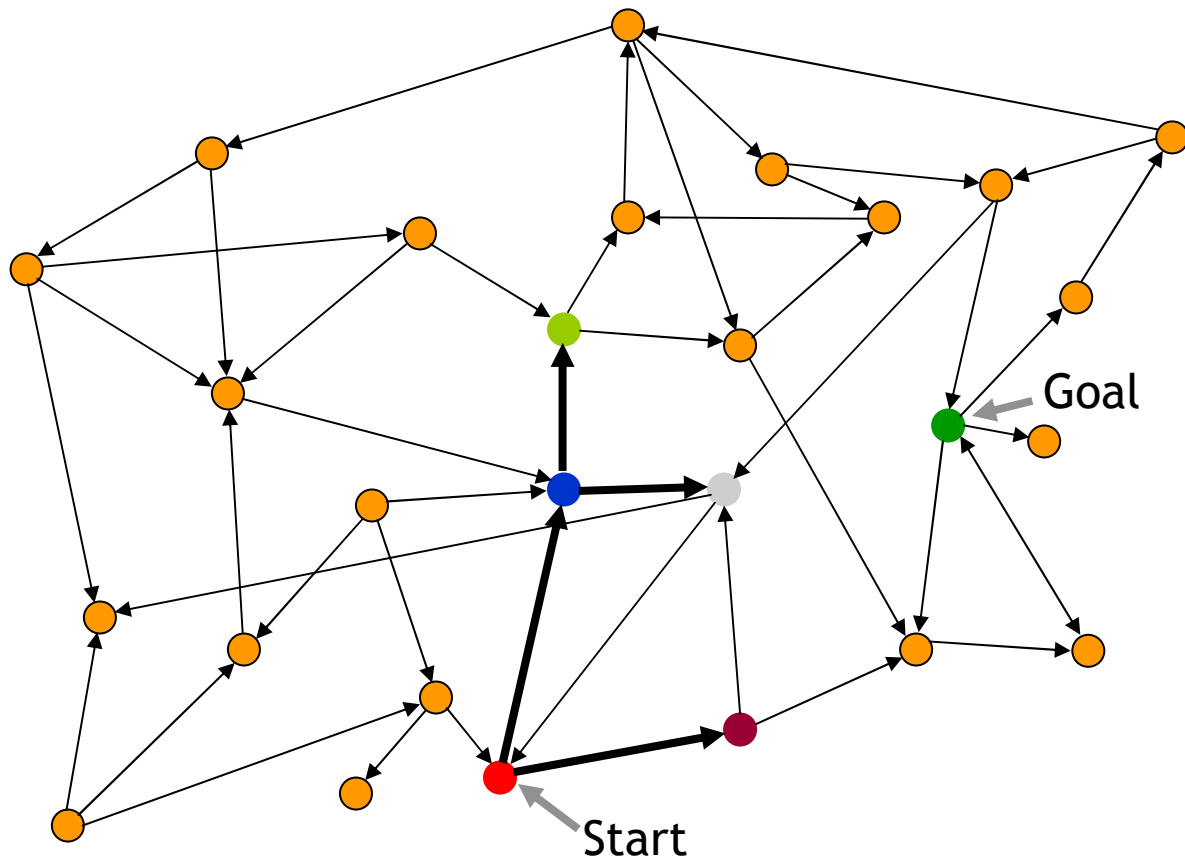
Ricerca nello spazio degli stati



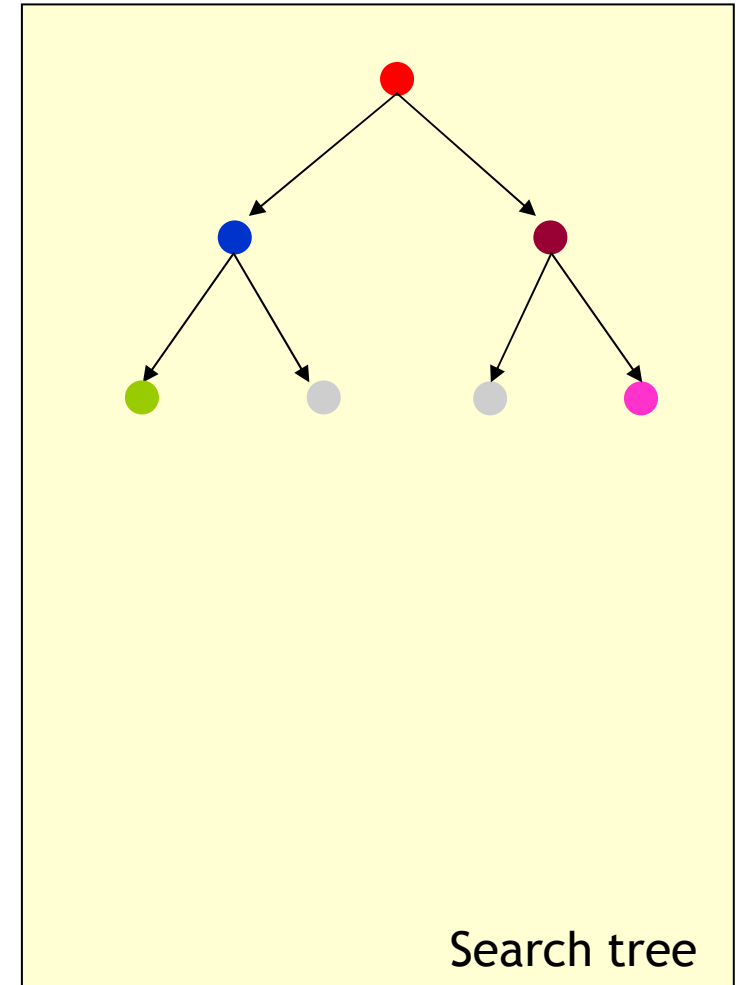
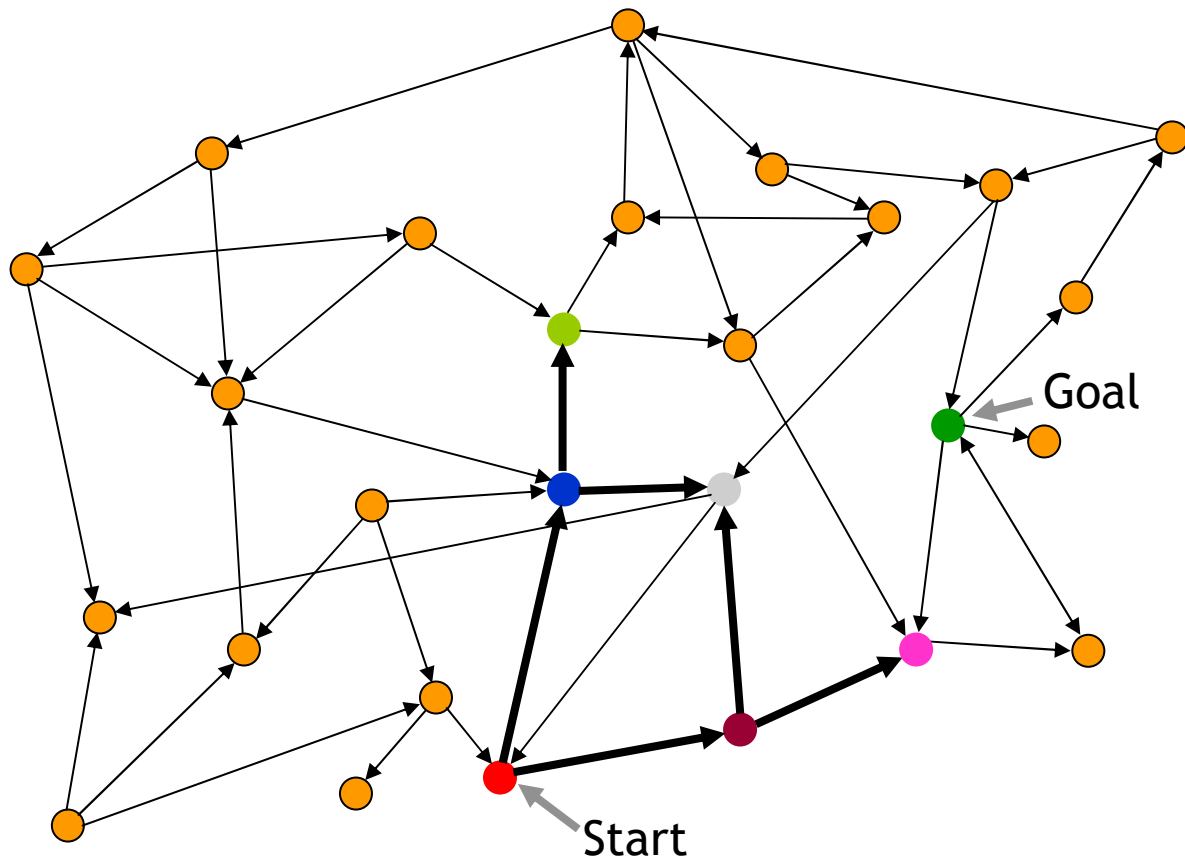
Ricerca nello spazio degli stati



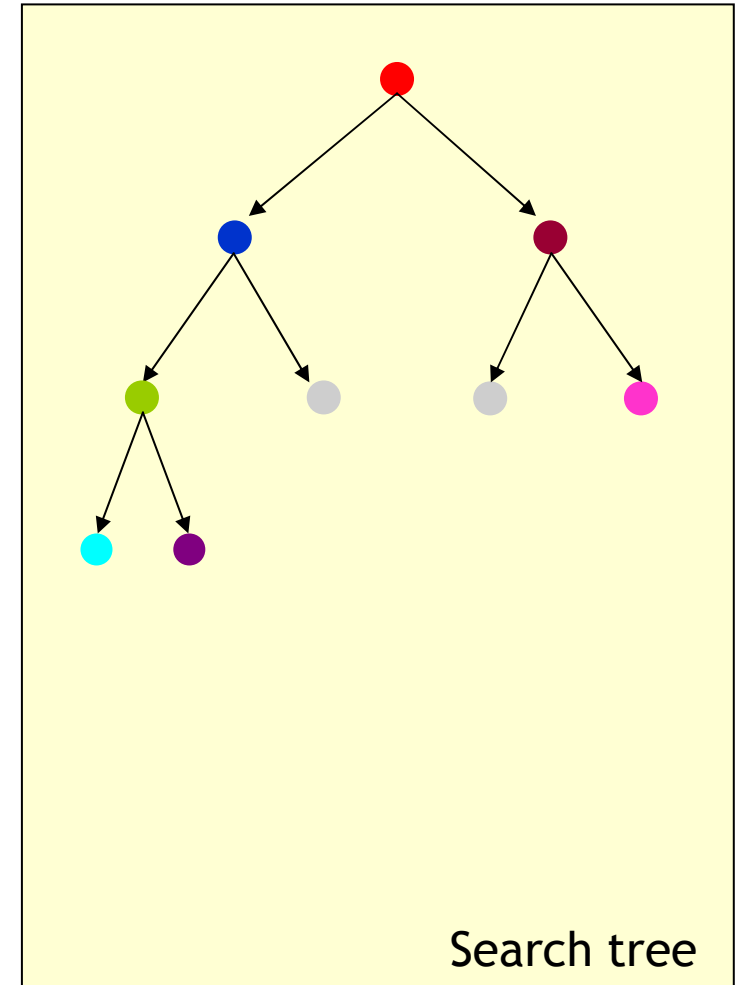
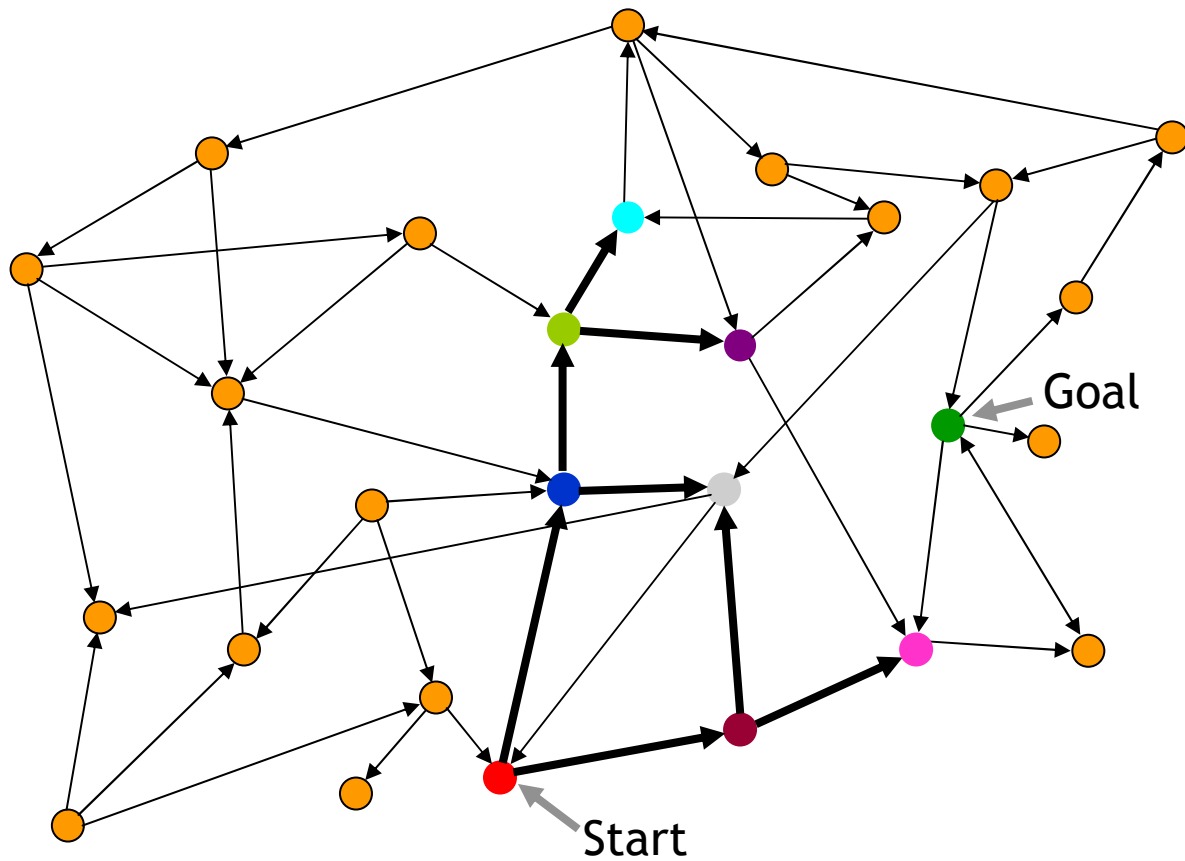
Ricerca nello spazio degli stati



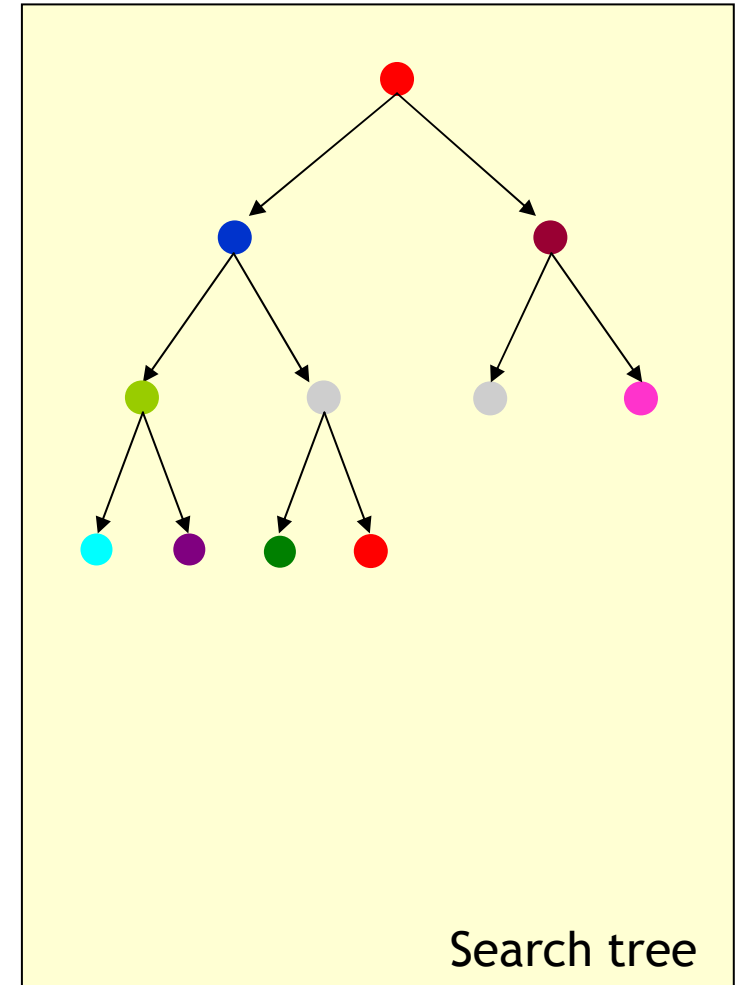
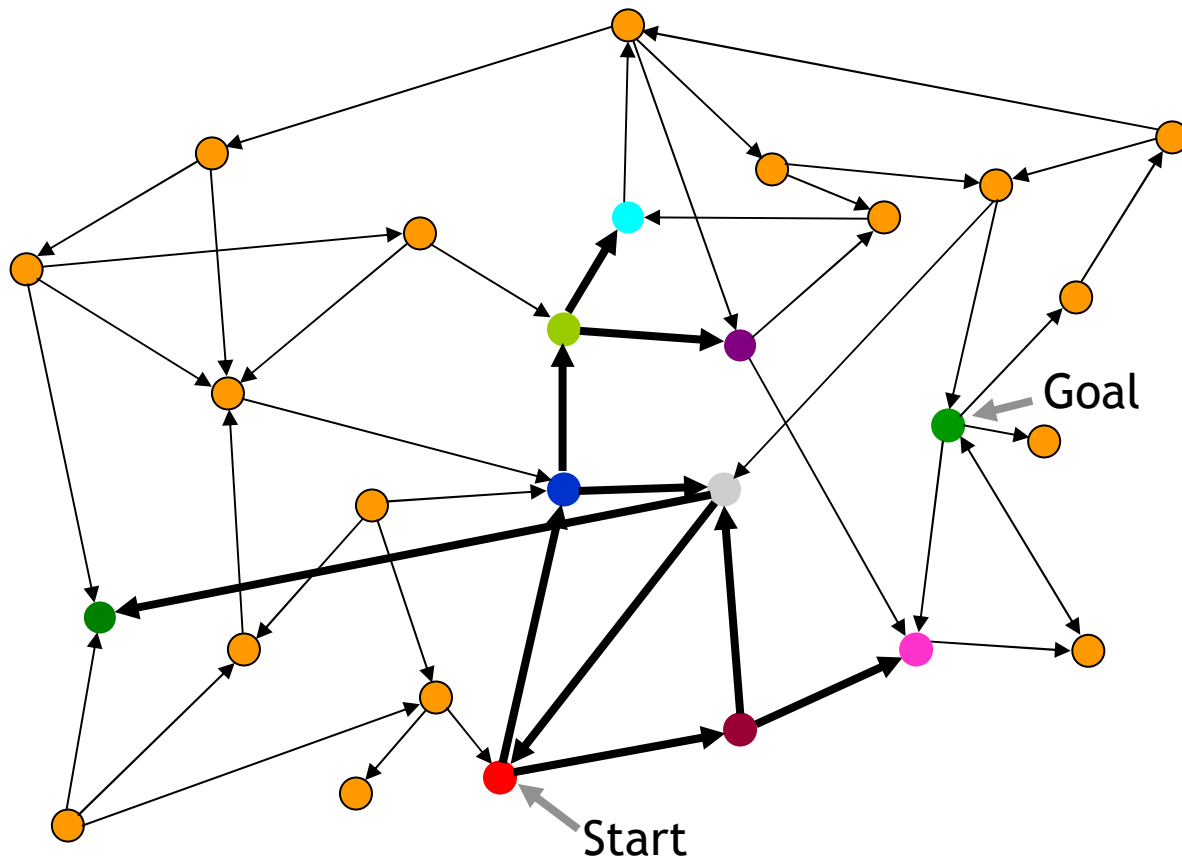
Ricerca nello spazio degli stati



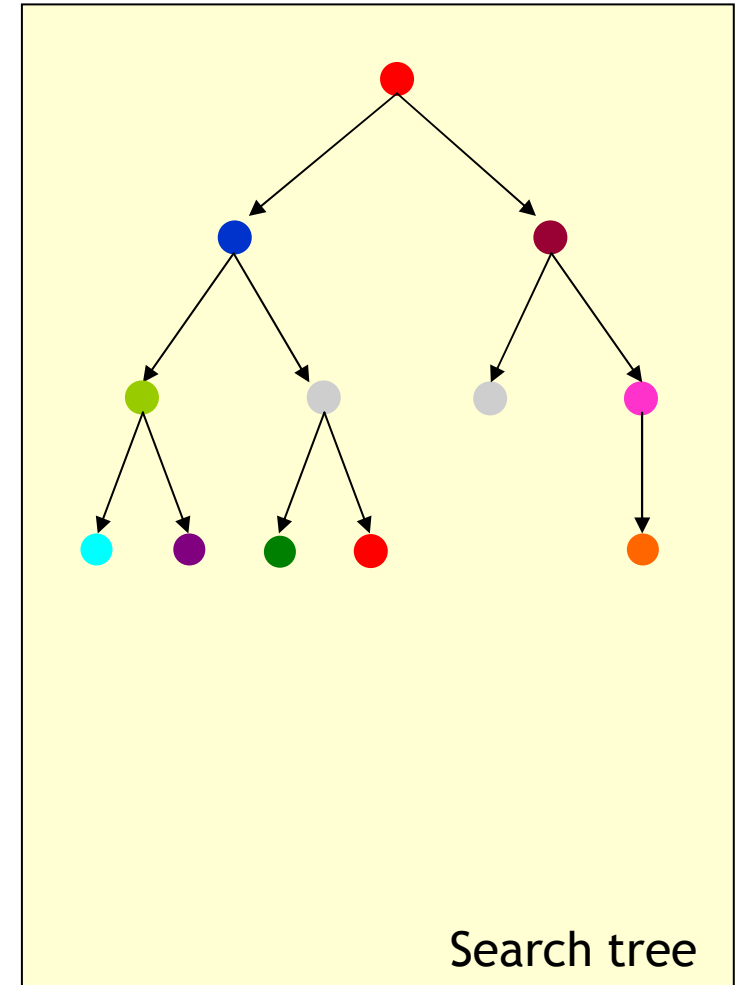
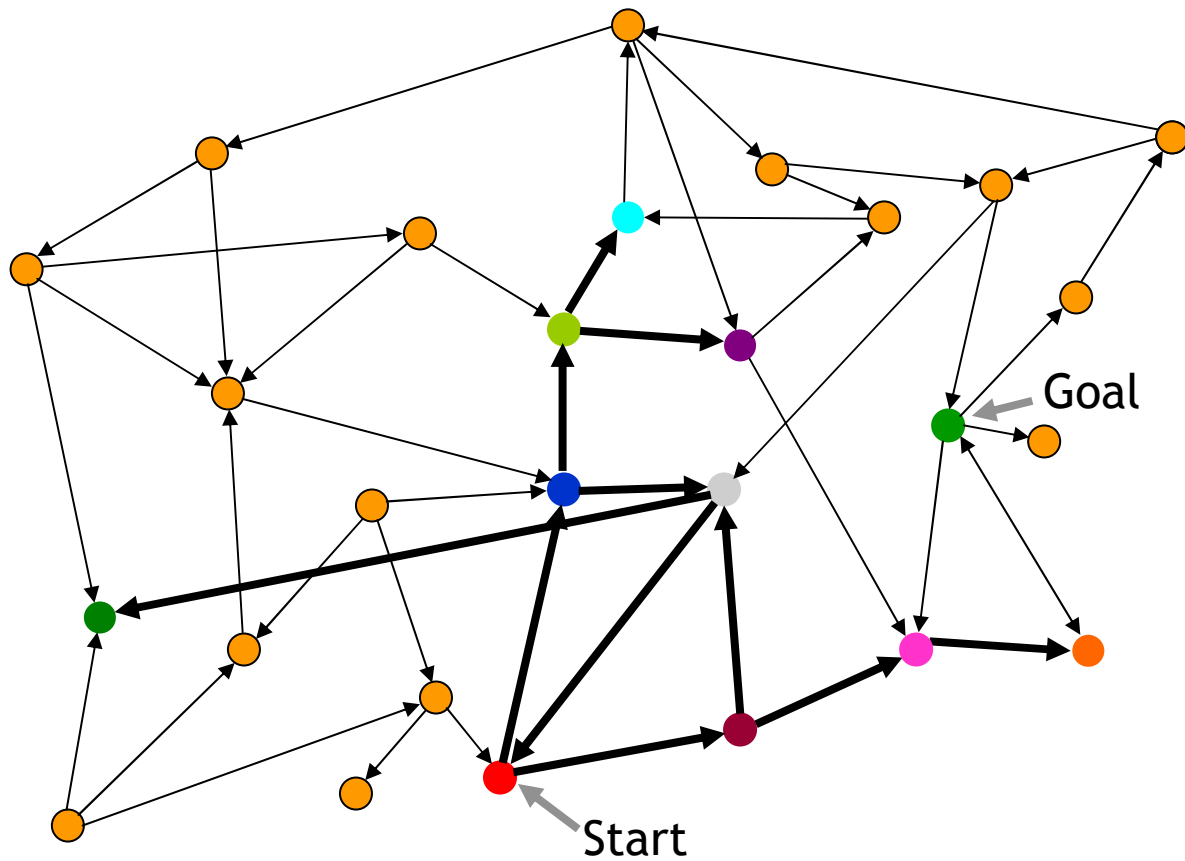
Ricerca nello spazio degli stati



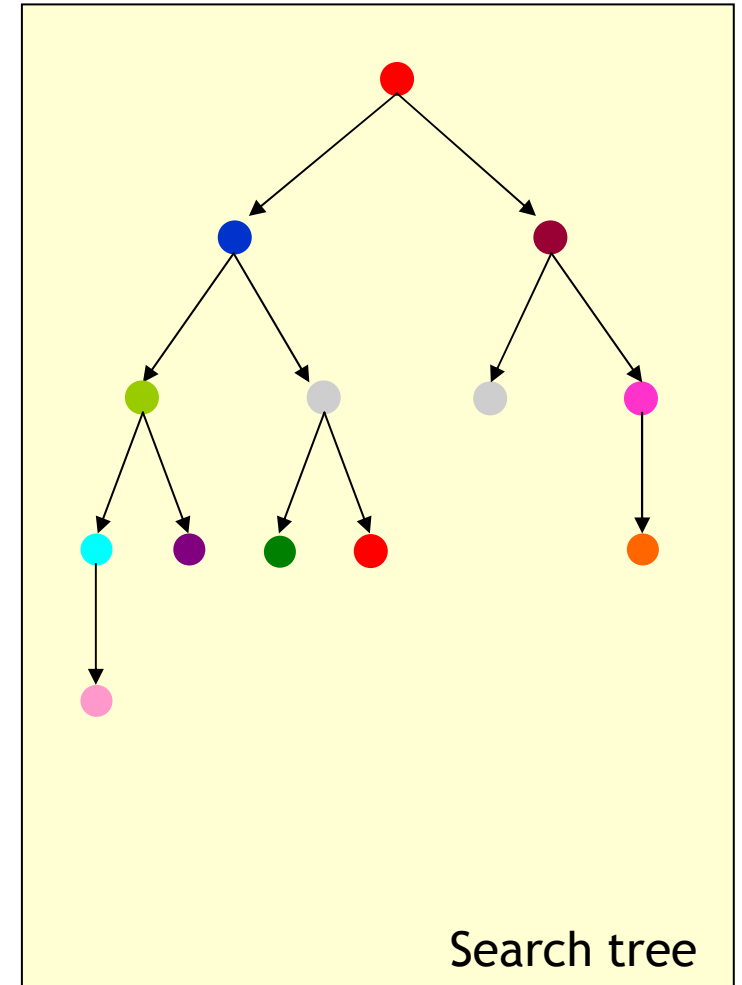
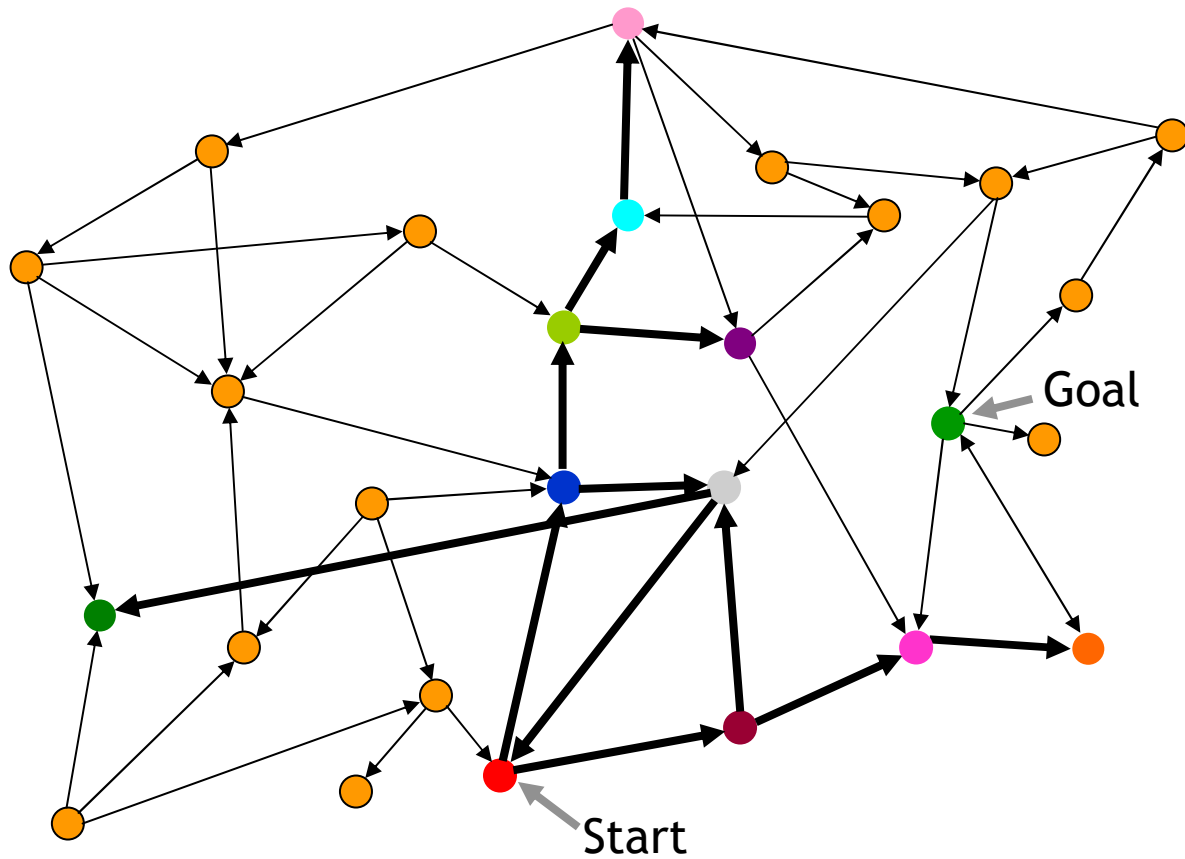
Ricerca nello spazio degli stati



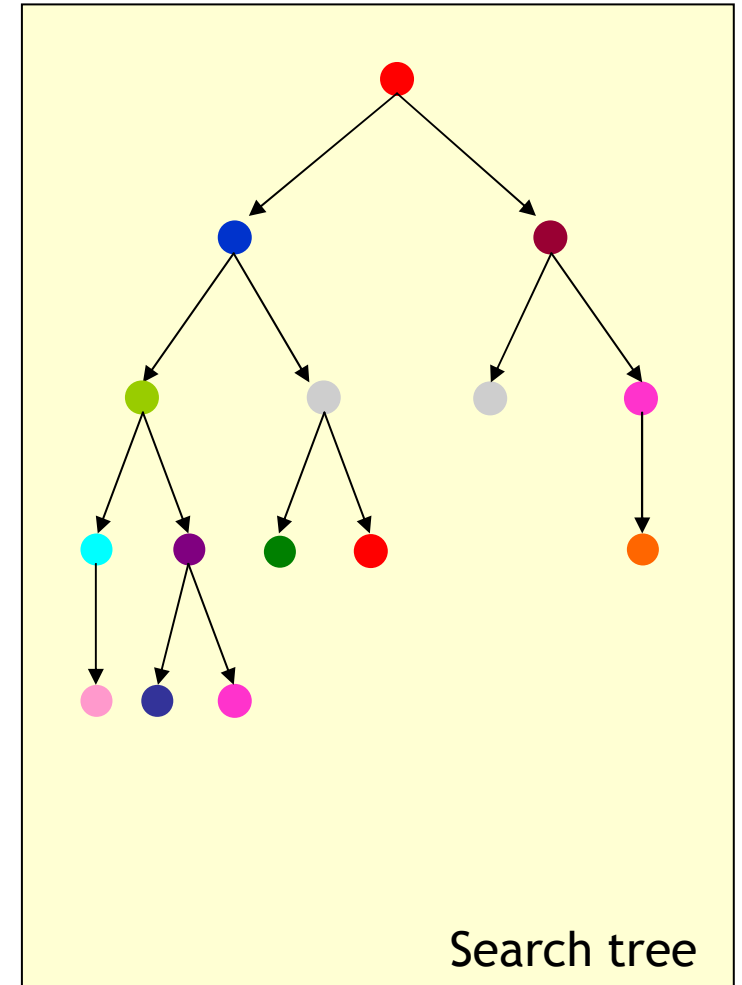
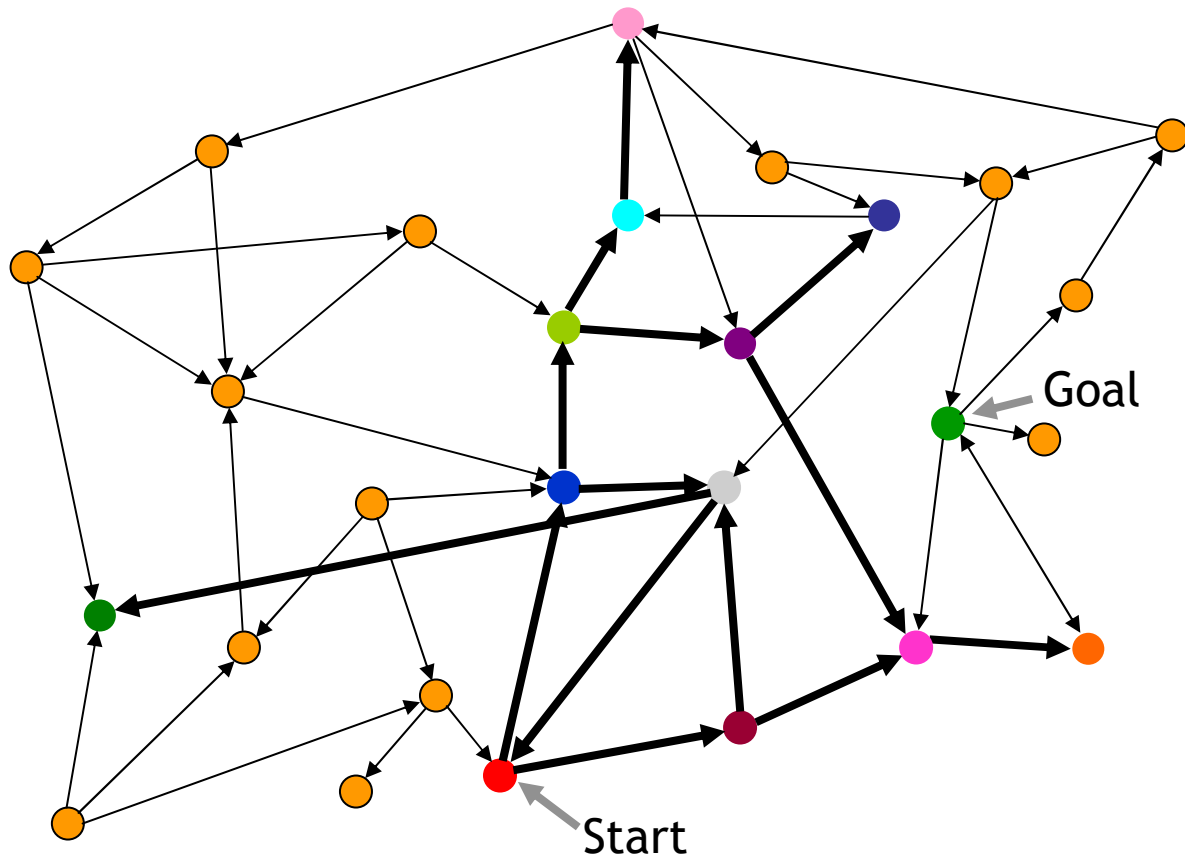
Ricerca nello spazio degli stati



Ricerca nello spazio degli stati

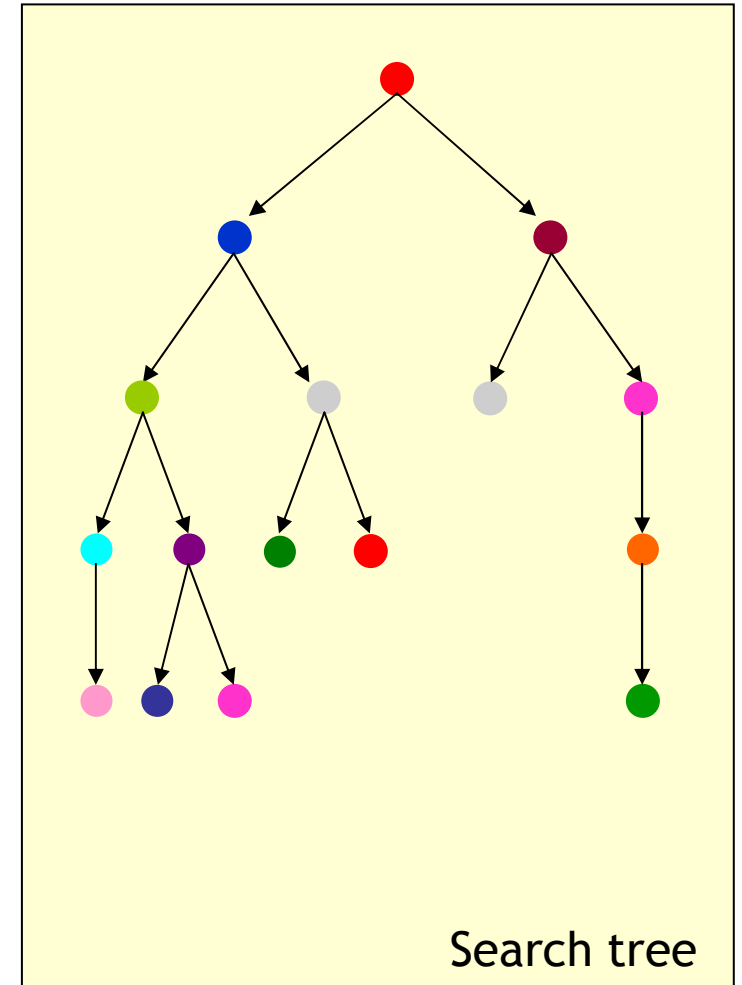
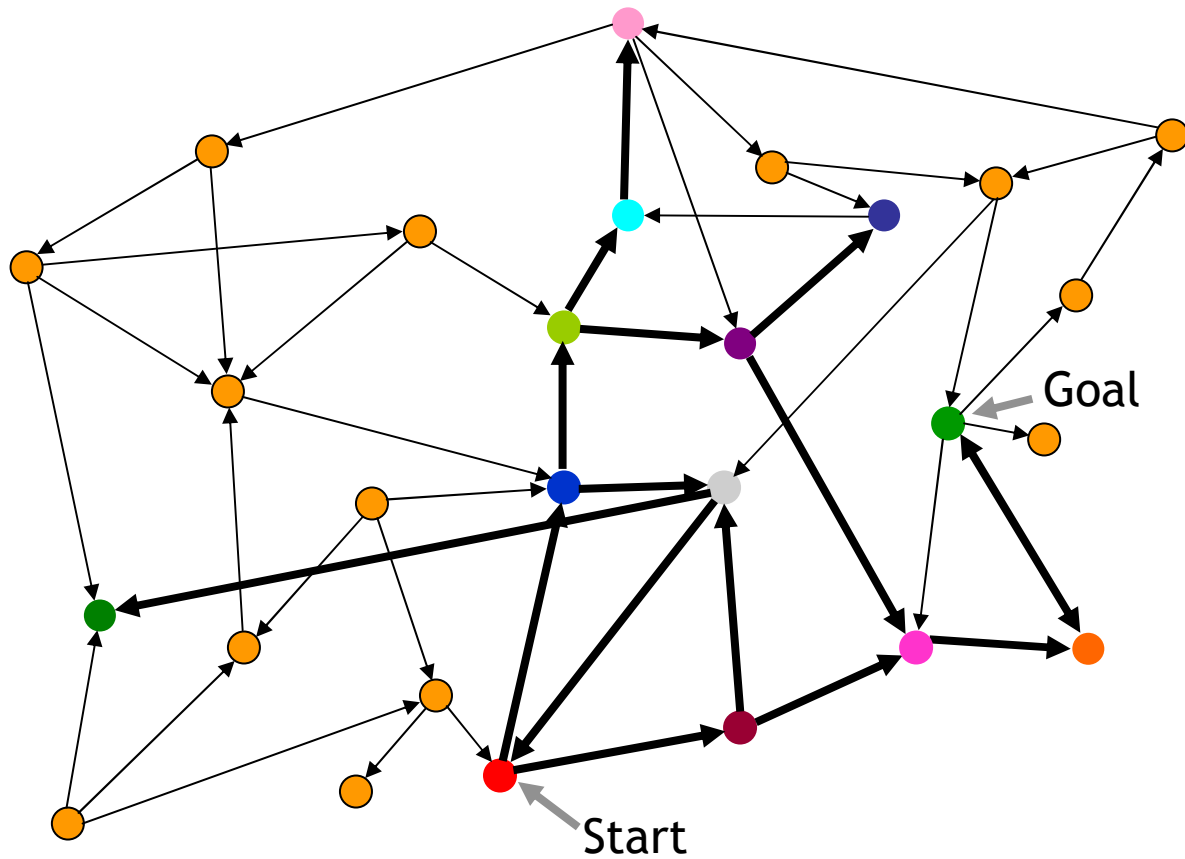


Ricerca nello spazio degli stati

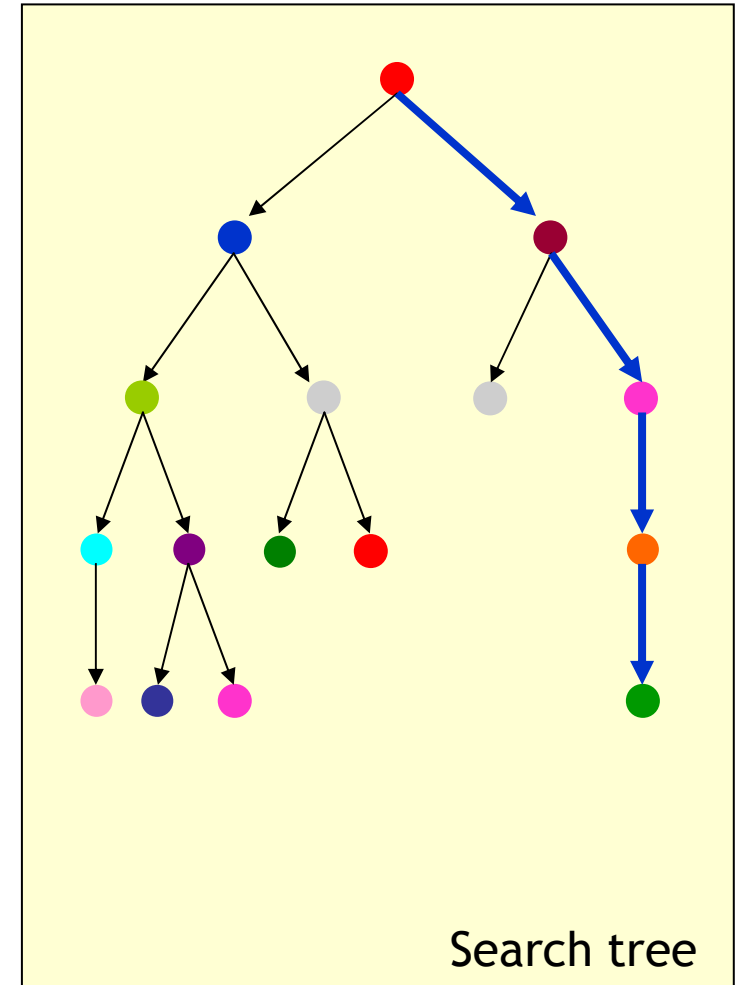
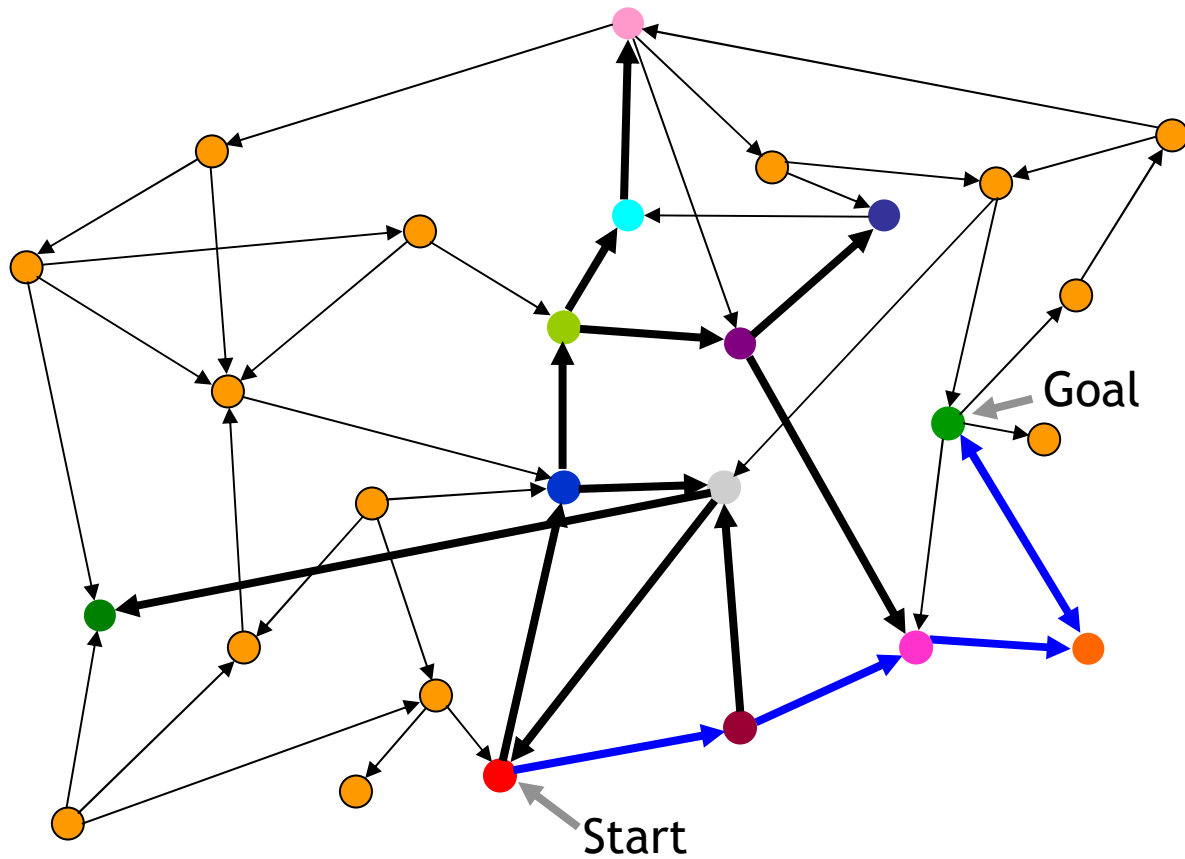


Search tree

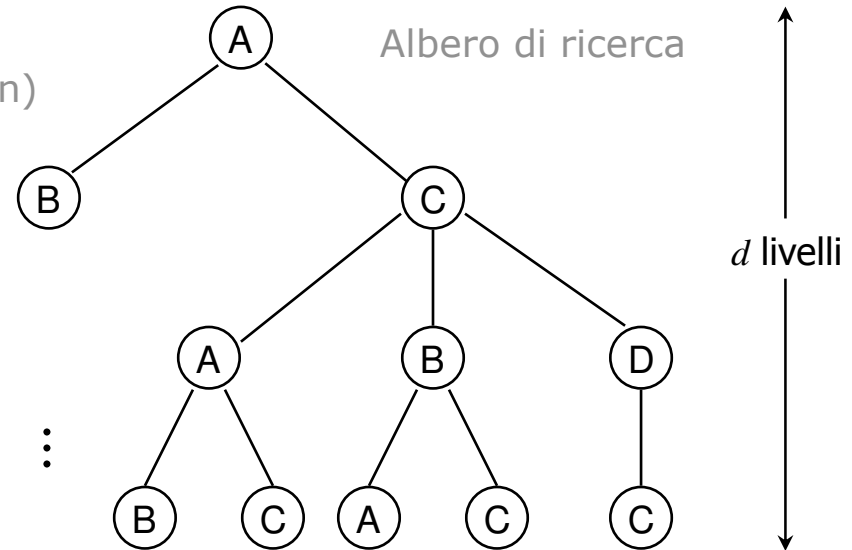
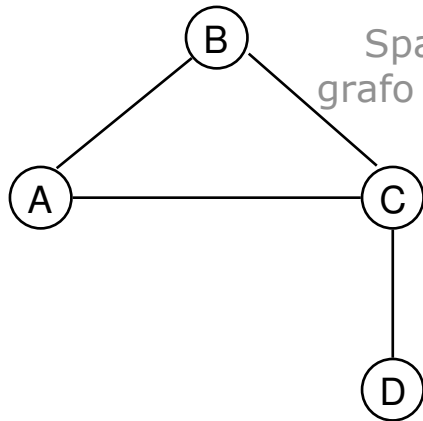
Ricerca nello spazio degli stati



Ricerca nello spazio degli stati



Definizioni



- Albero di ricerca

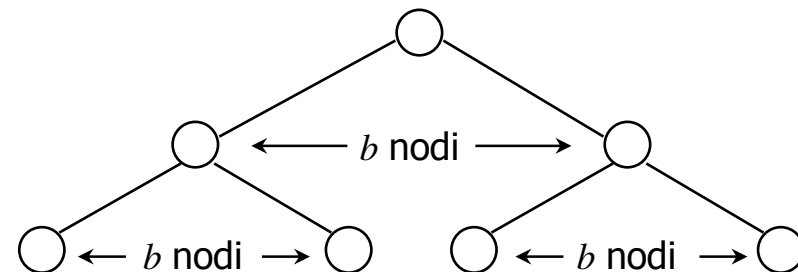
Nodo radice (stato iniziale)

Branching factor (b)

Numero (massimo) di nodi figli

Profondità (d)

Numero di livelli



Strategie di ricerca

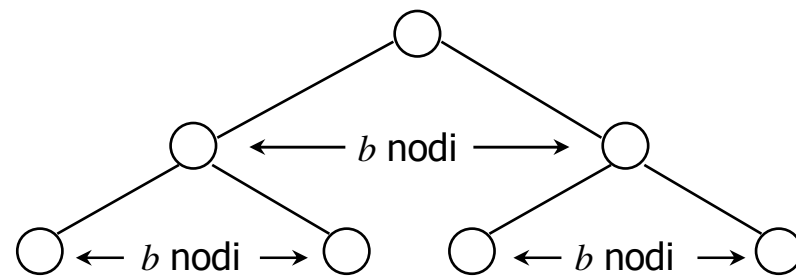
▪ Parametri di valutazione

Completezza: è certo che si trovi una soluzione?
(o tutte le soluzioni, se necessario)

Ottimalità: in presenza di diverse soluzioni, verrà trovata la soluzione migliore?

Complessità di tempo: quanti passi saranno necessari?

Complessità di memoria: quanto grande sarà la struttura in memoria?



Numero di nodi di un albero con branching factor costante:

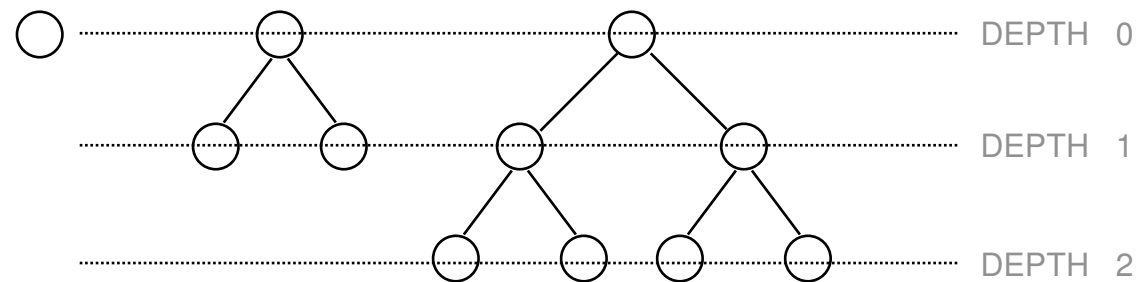
$$n = 1 + b + b^2 + b^3 + \dots + b^d$$

Strategie di ricerca: *breadth first*

- Ricerca esaustiva in ampiezza

Si espandono tutti i nodi di un livello prima di passare al successivo

Per la scelta dei nodi da espandere si usa una strategia FIFO (una coda o *queue*)



Completa

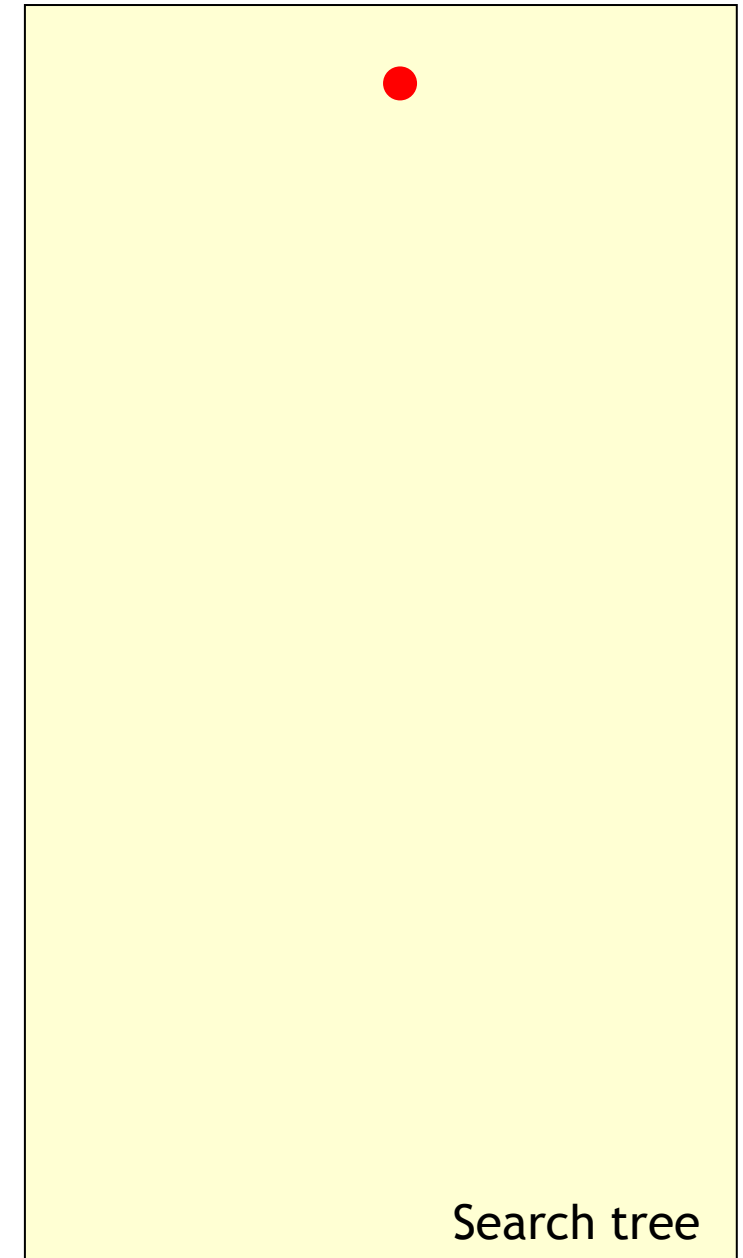
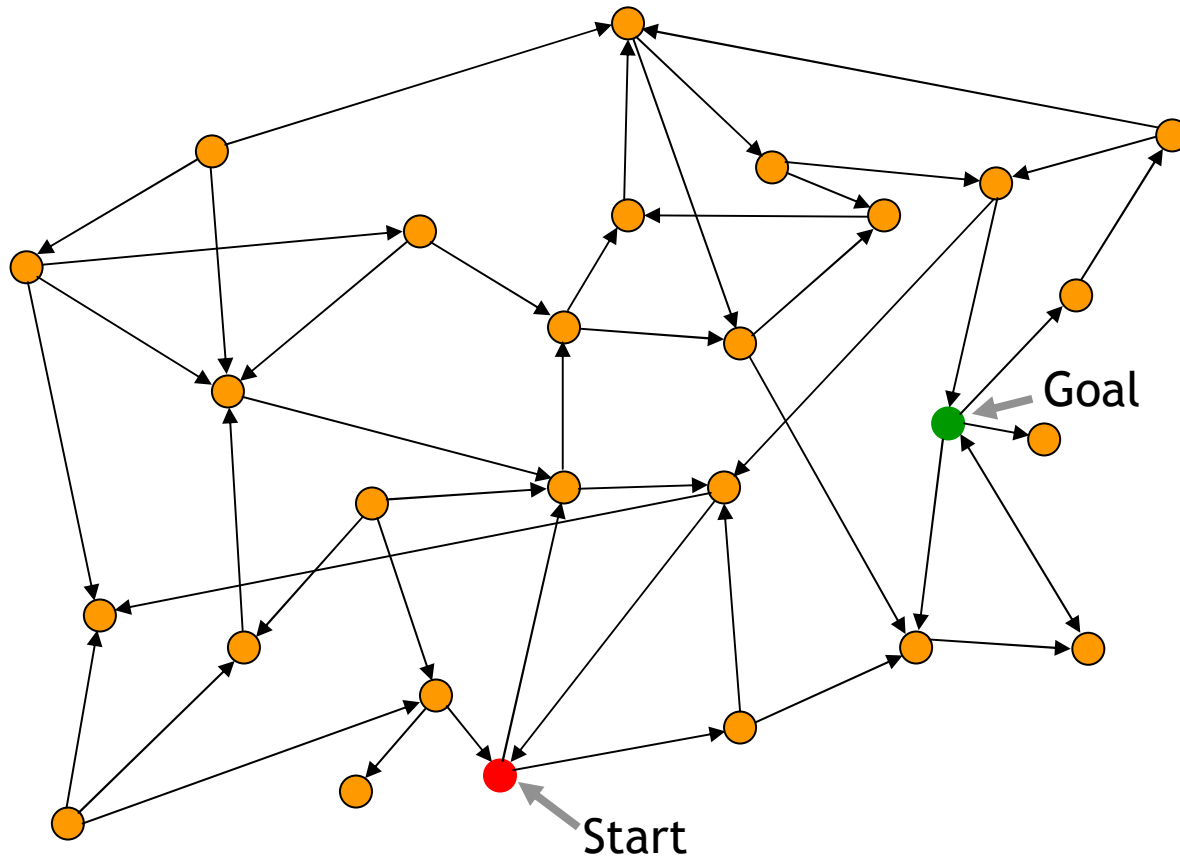
Ottimale

Complessità di tempo: $O(b^d)$

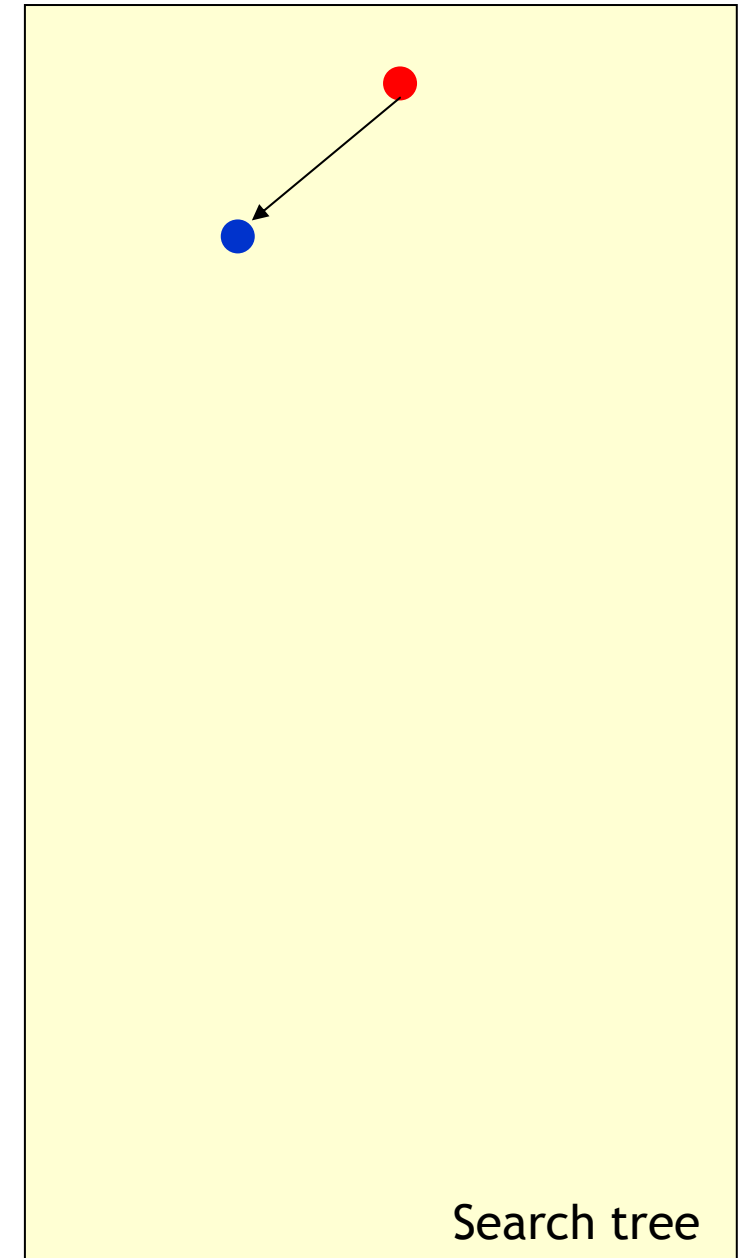
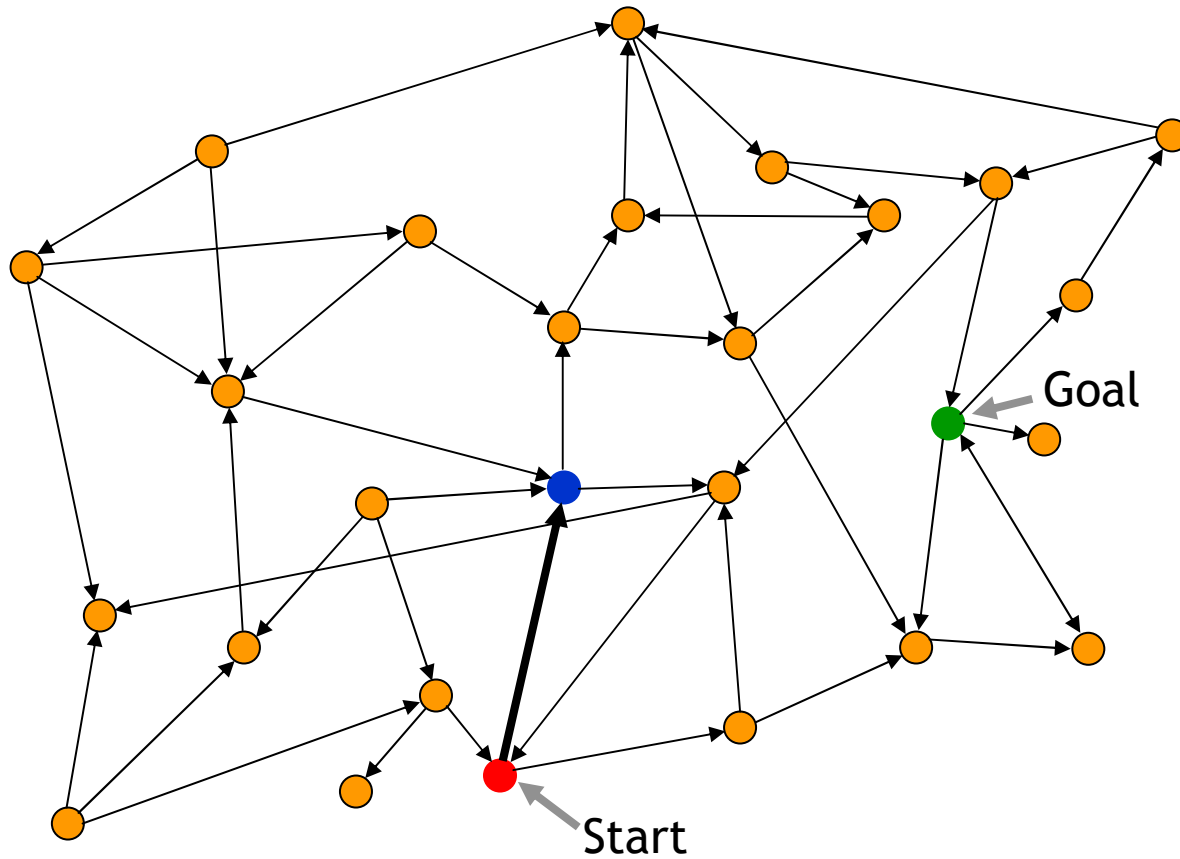
Complessità di memoria: $O(b^d)$

Uso: quando espandere tutto l'albero è necessario o conveniente

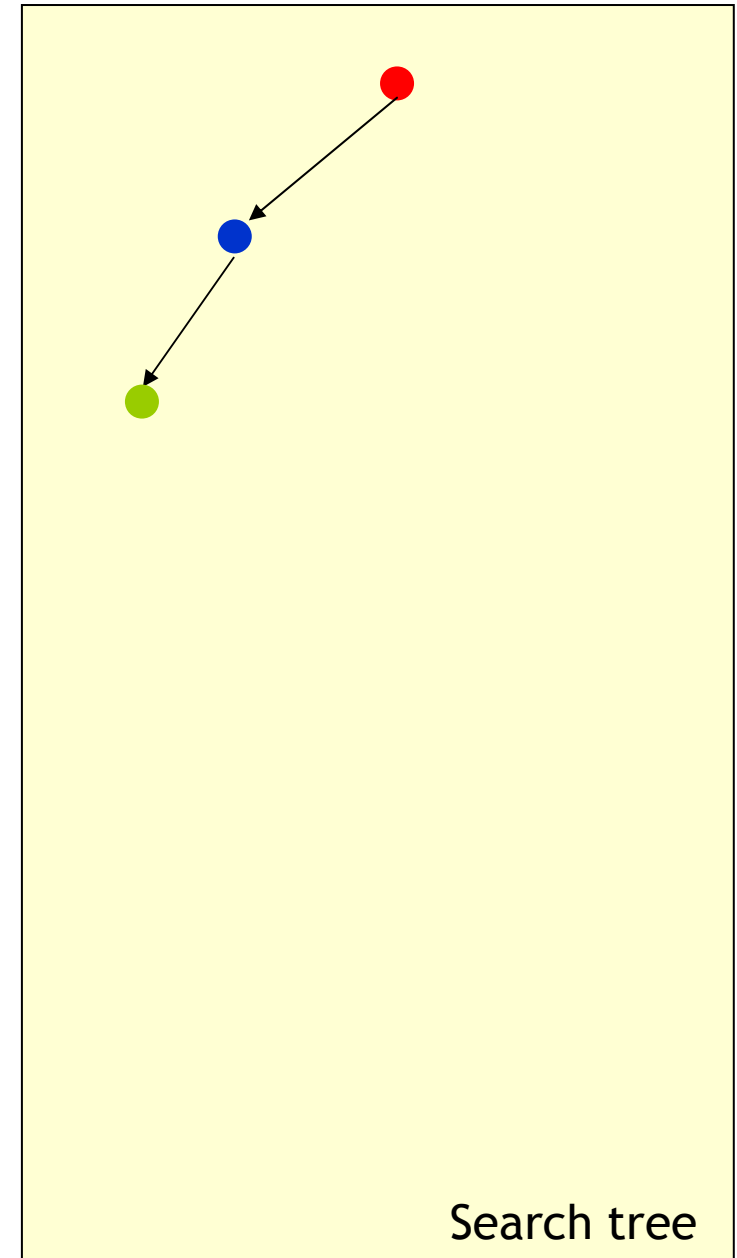
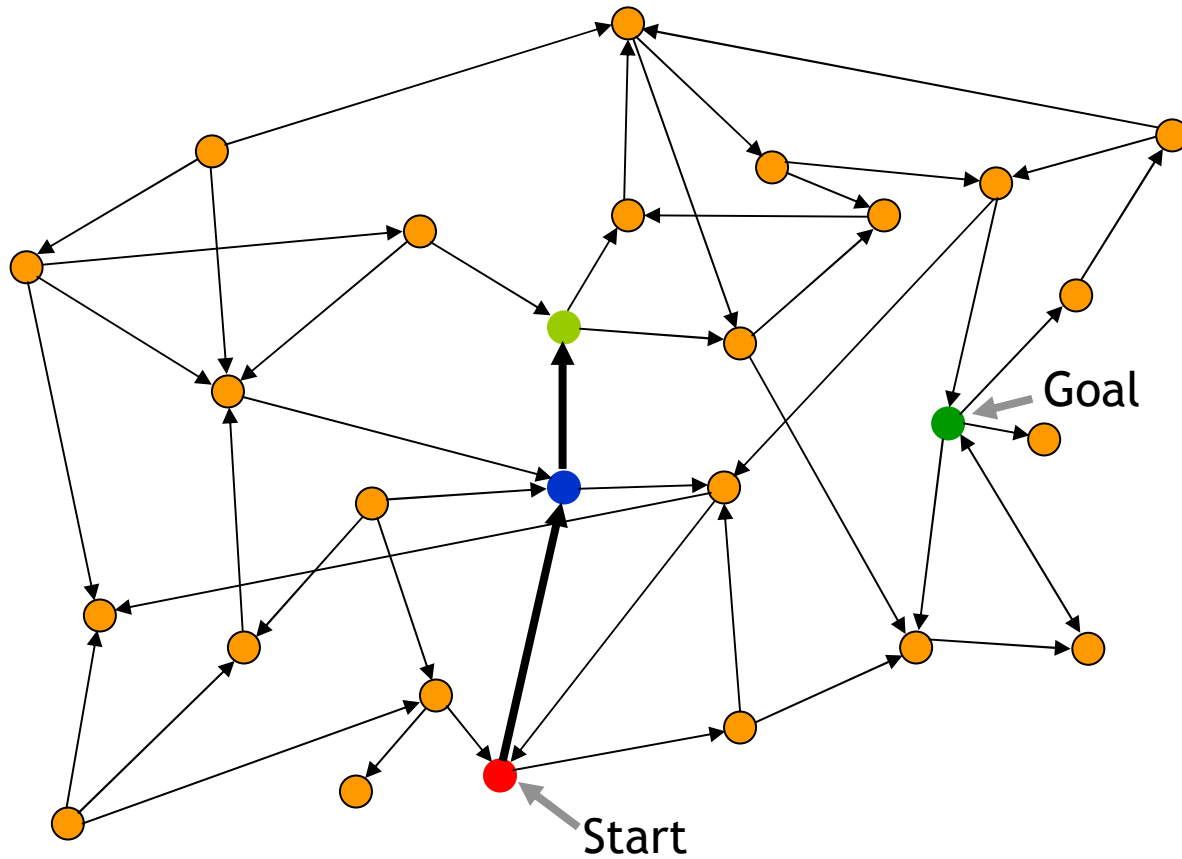
Ricerca nello spazio degli stati



Ricerca nello spazio degli stati

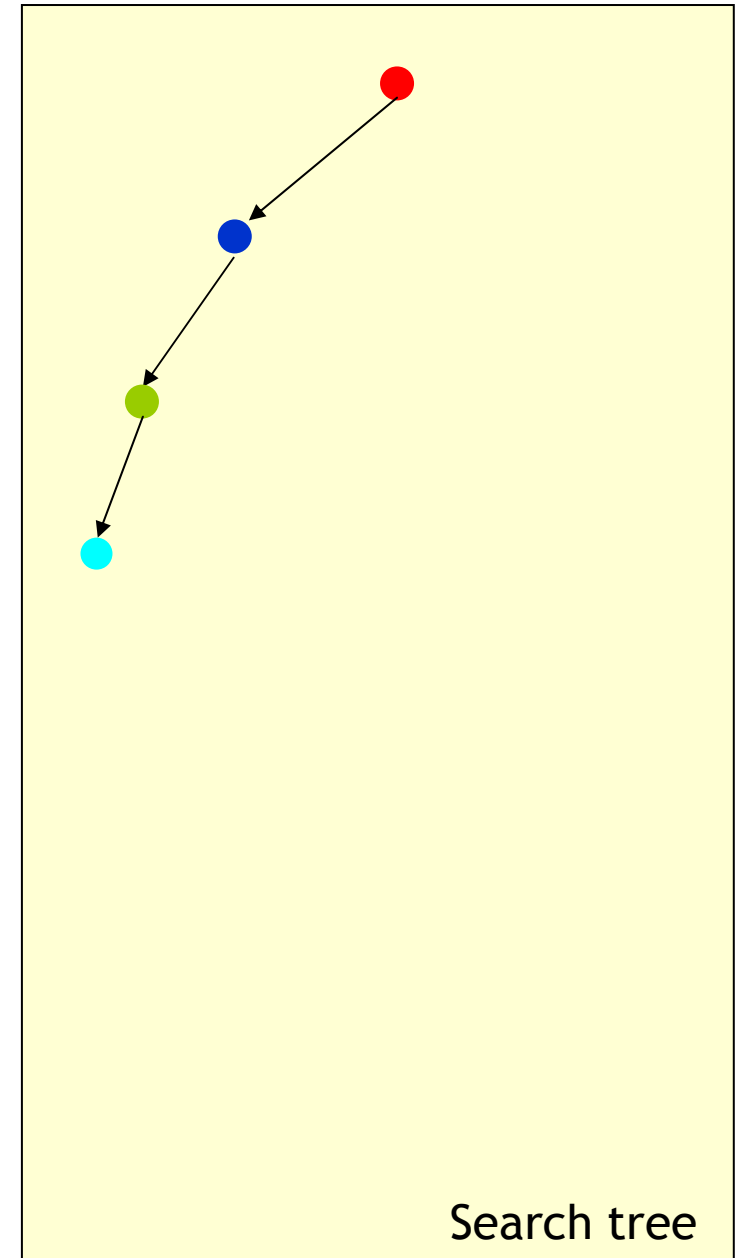
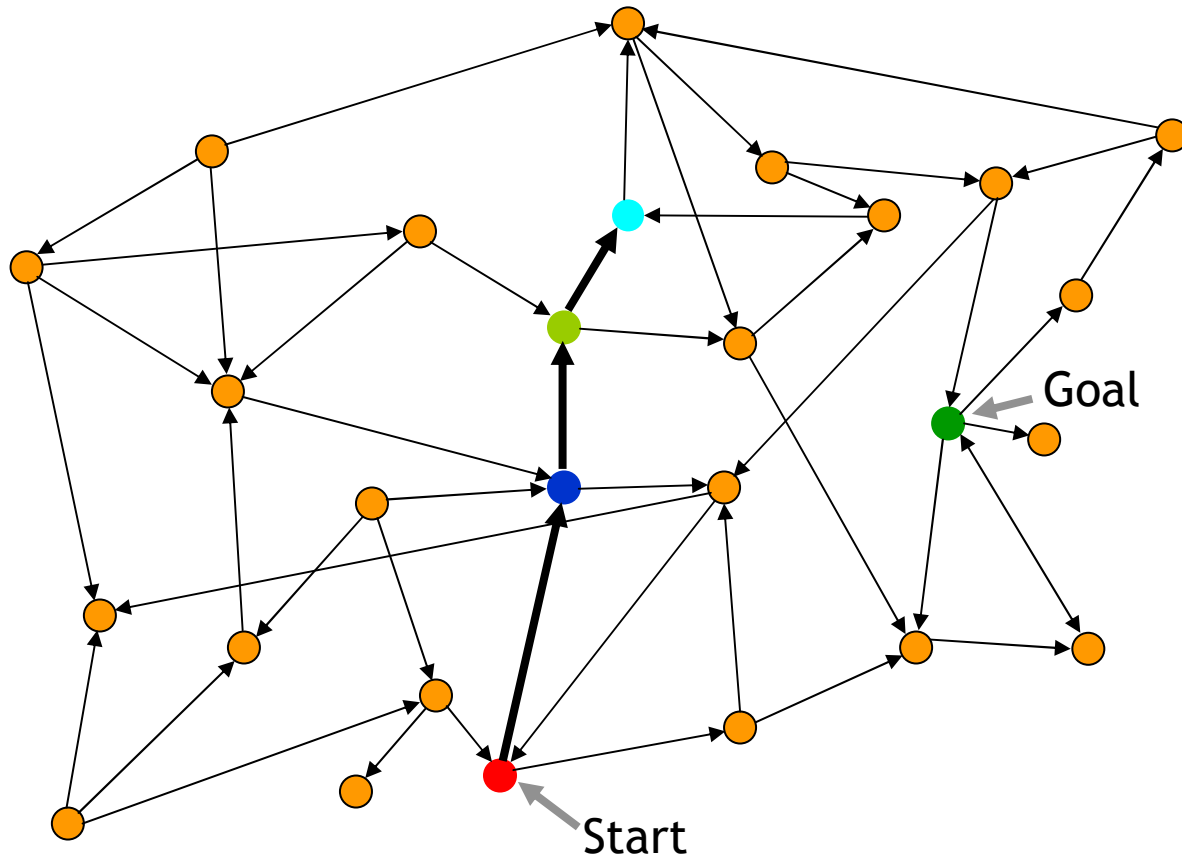


Ricerca nello spazio degli stati



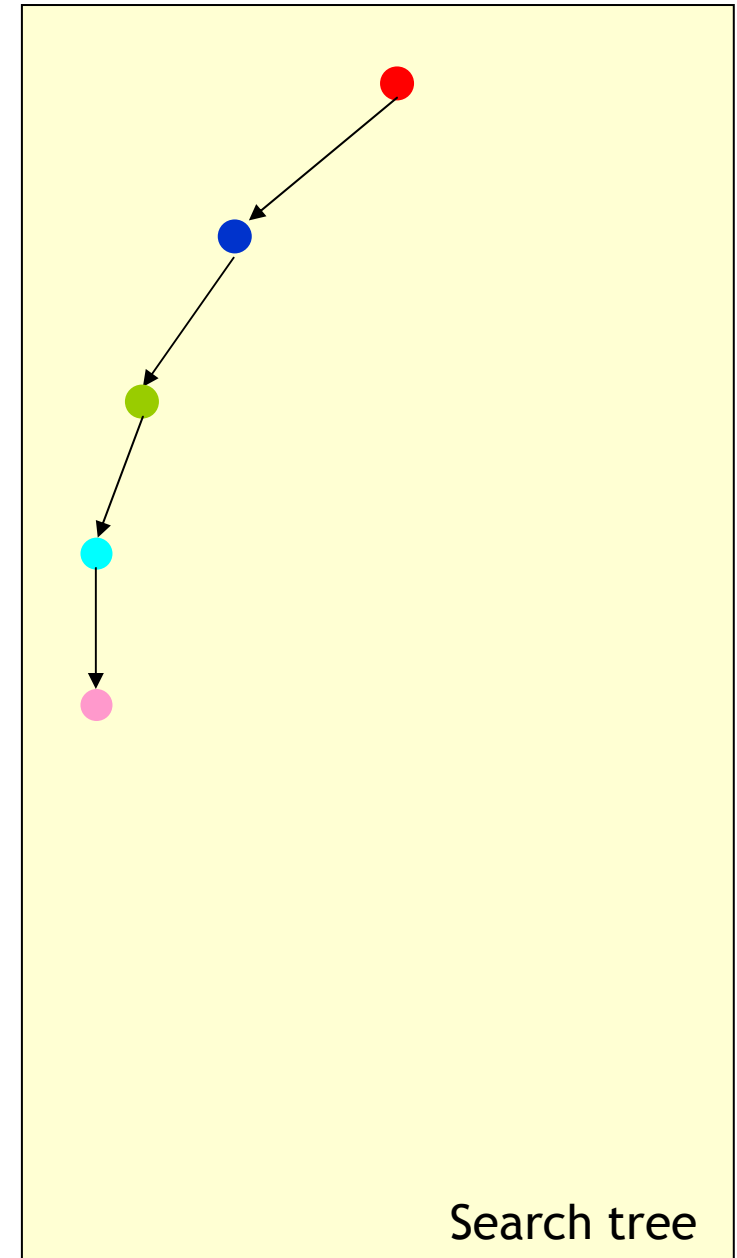
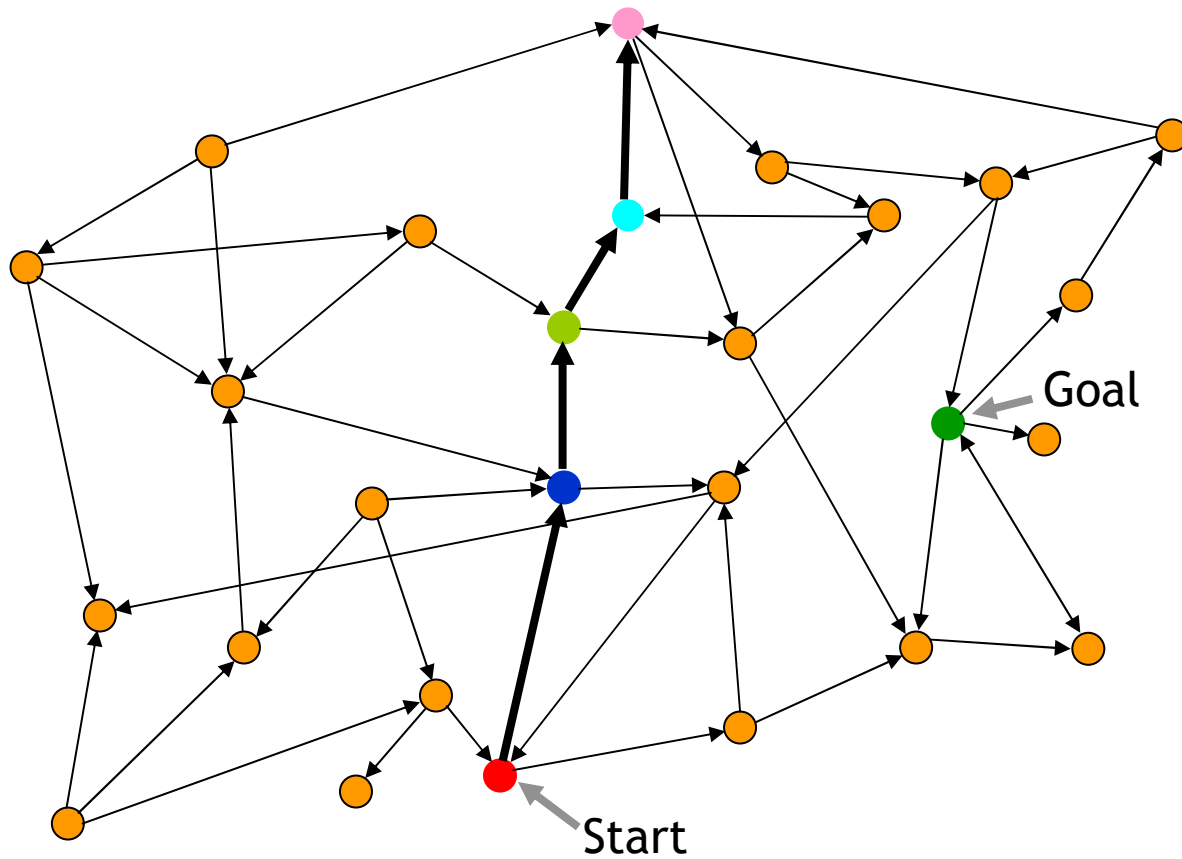
Search tree

Ricerca nello spazio degli stati

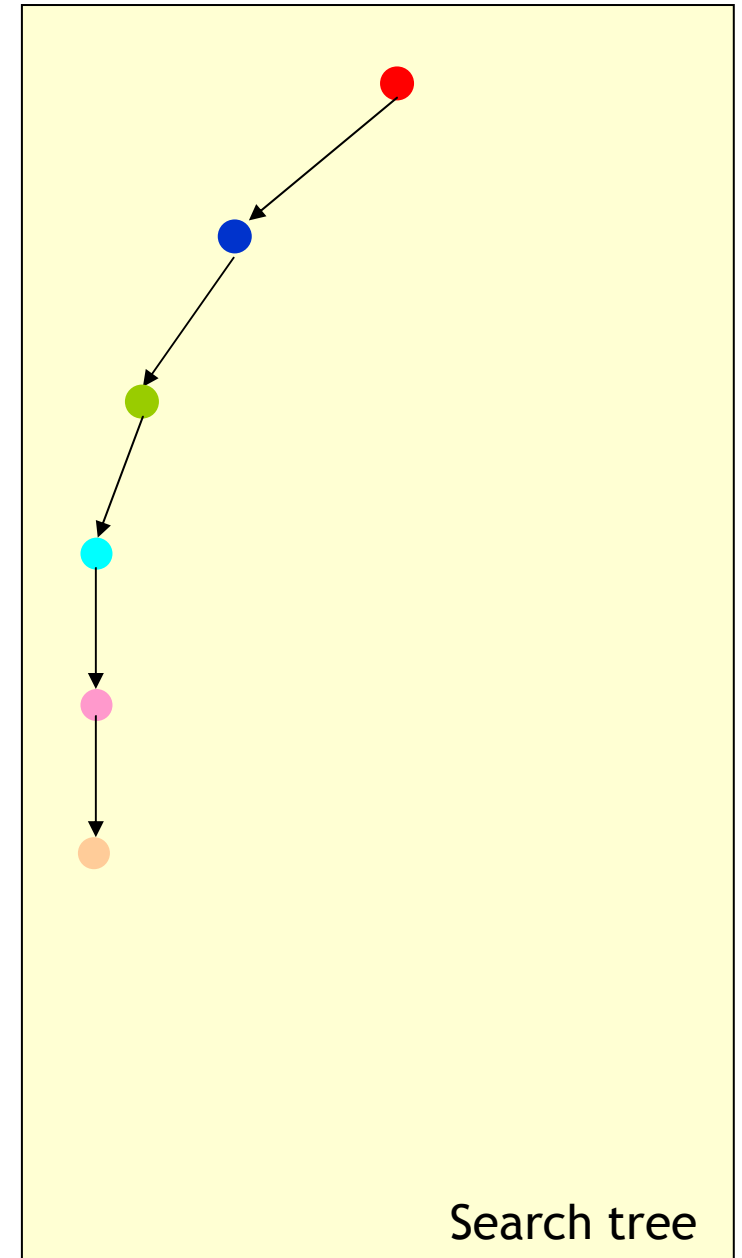
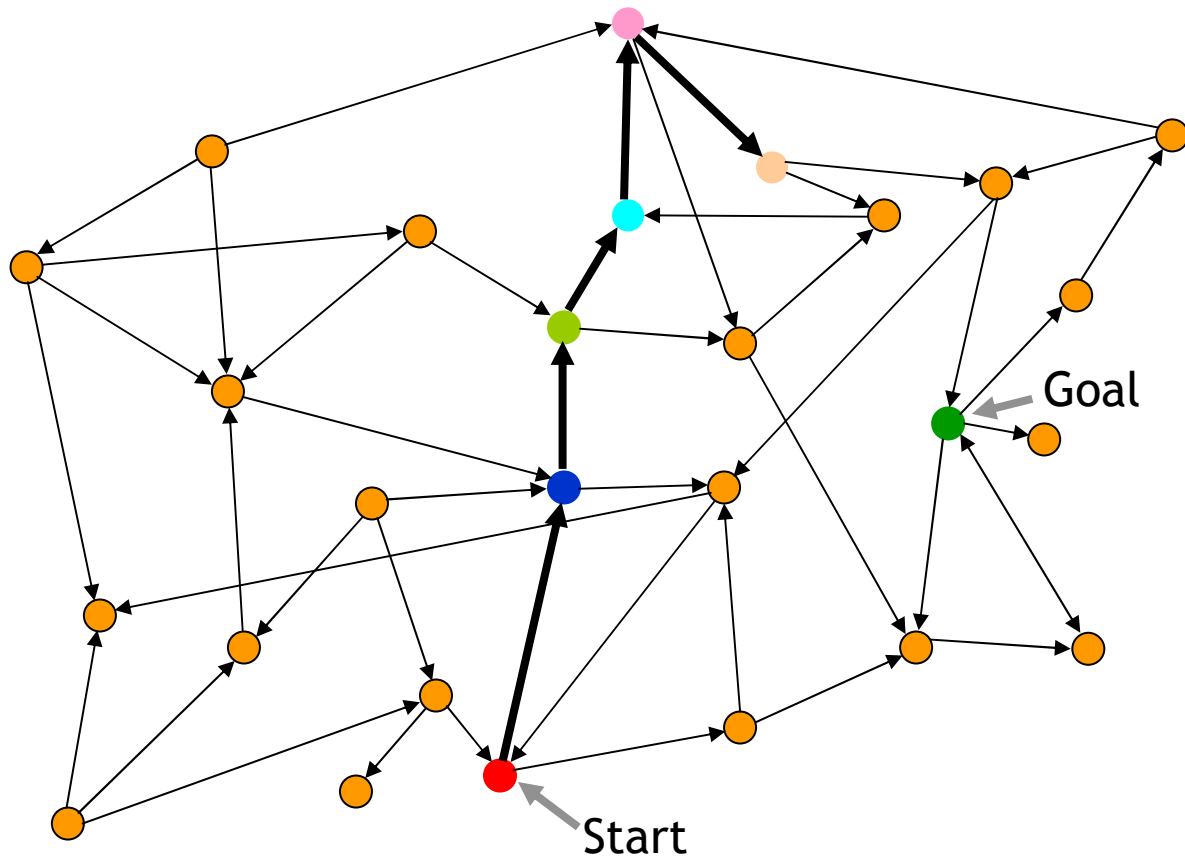


Search tree

Ricerca nello spazio degli stati

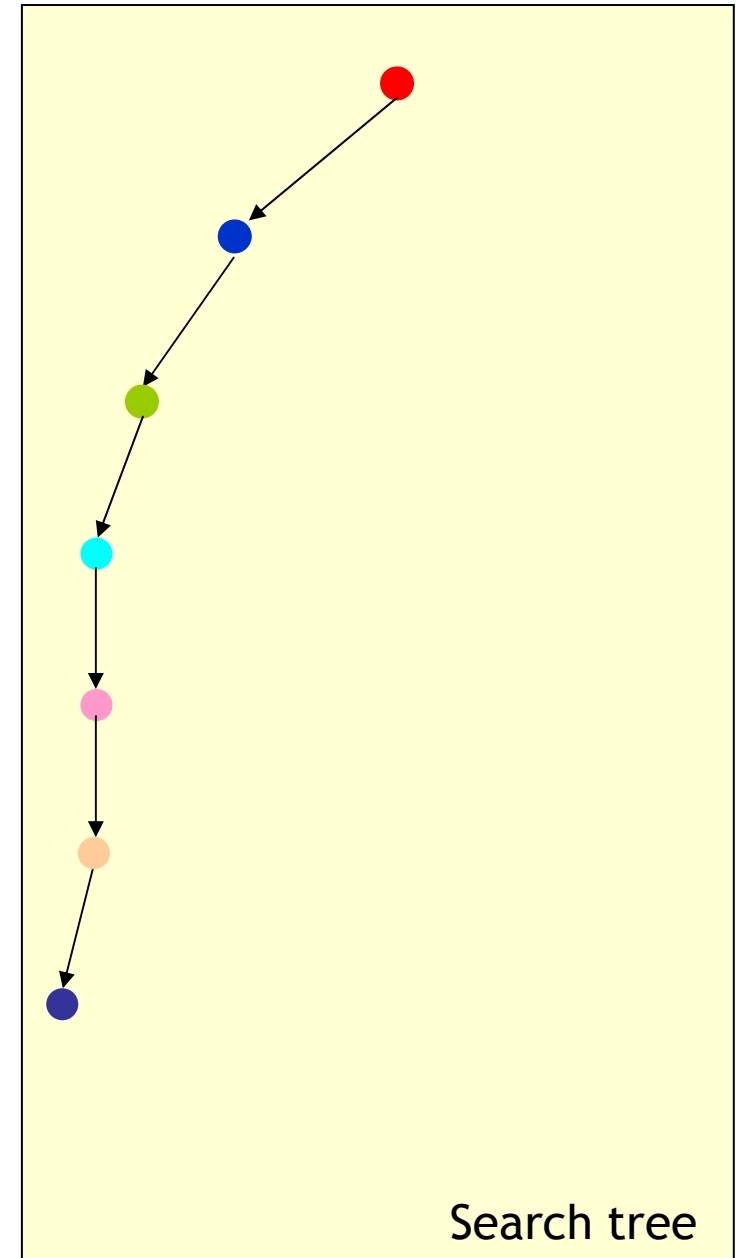
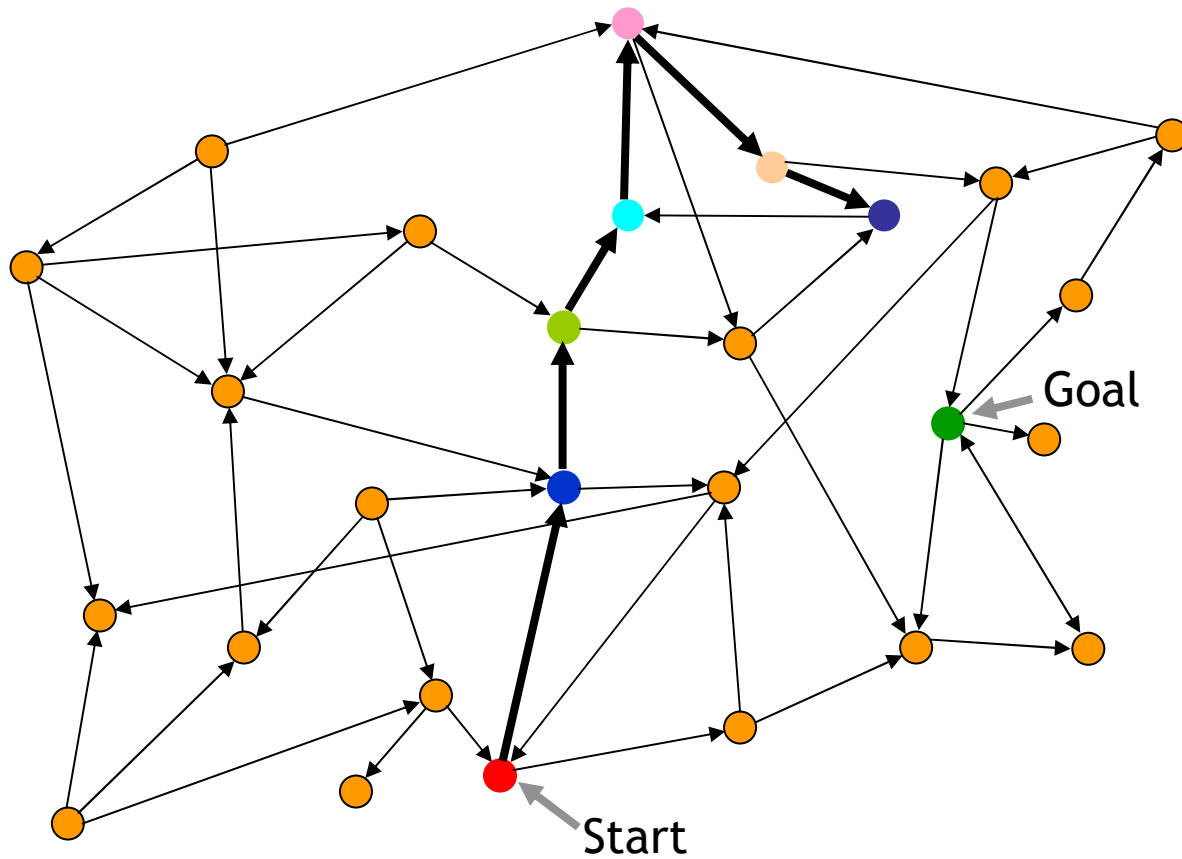


Ricerca nello spazio degli stati

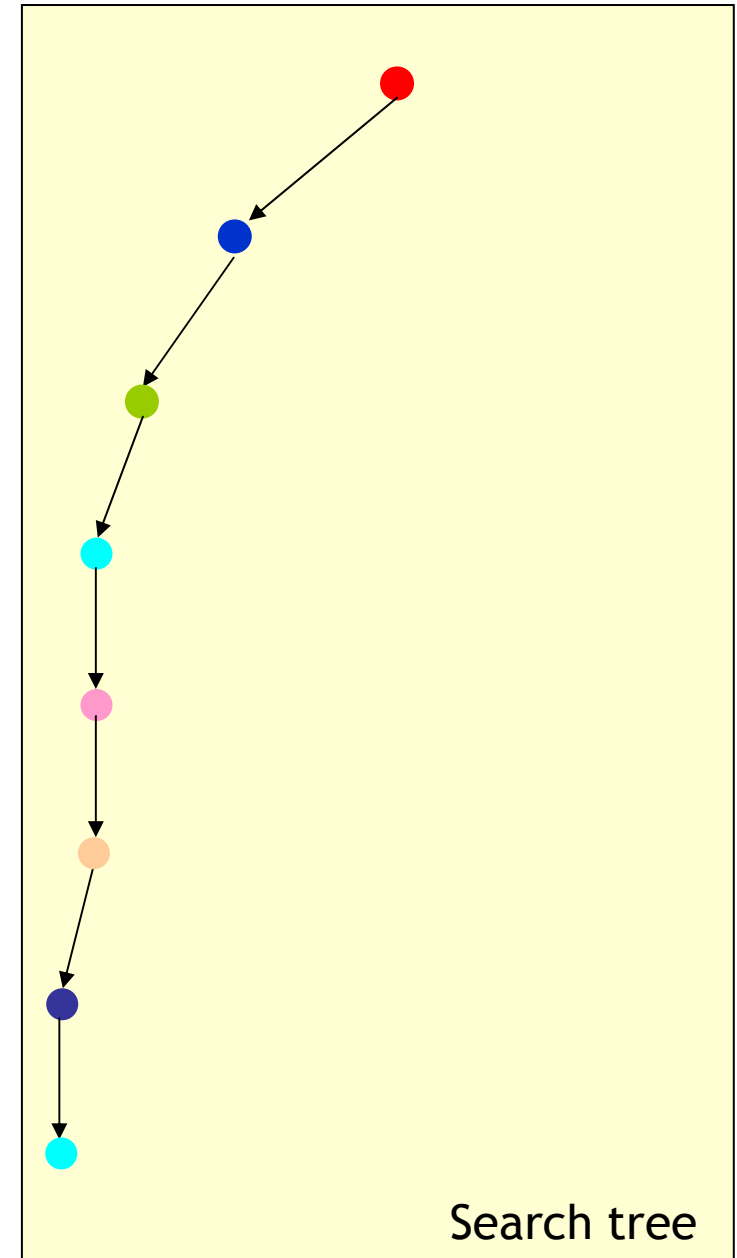
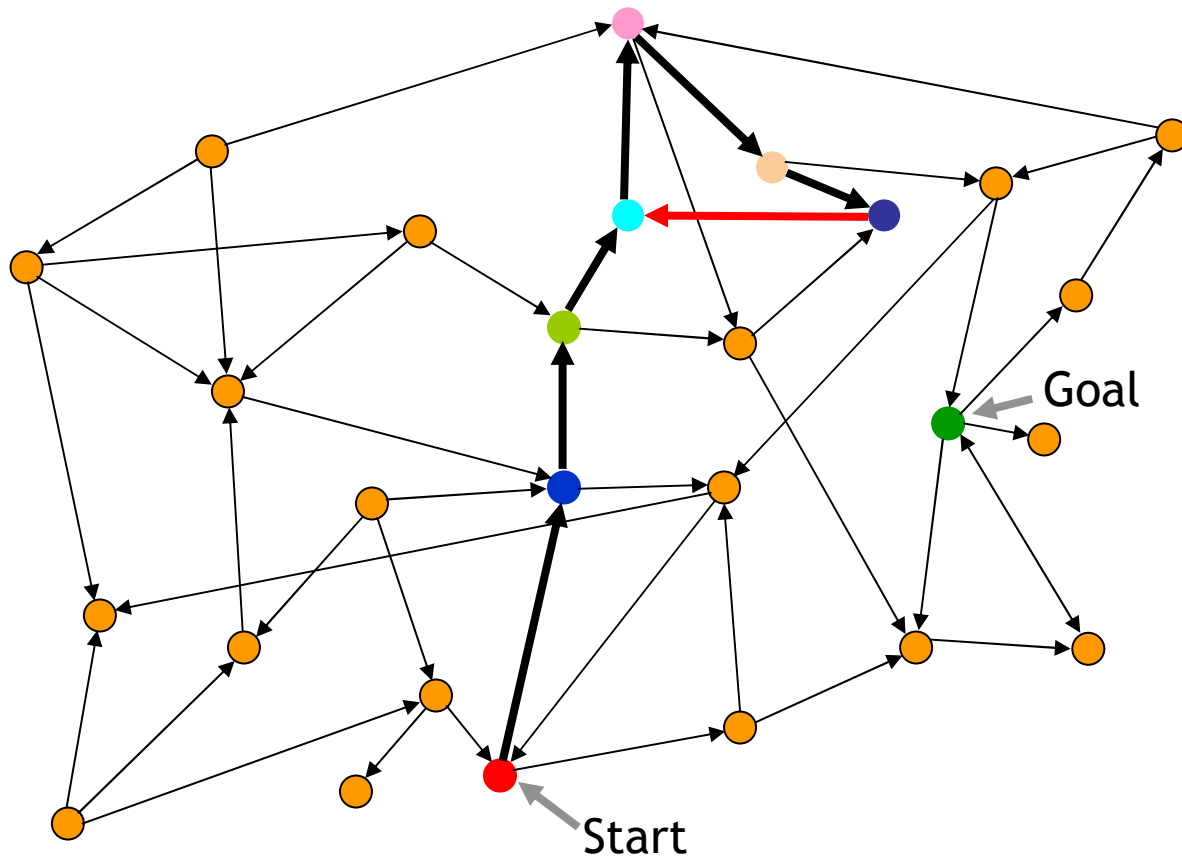


Search tree

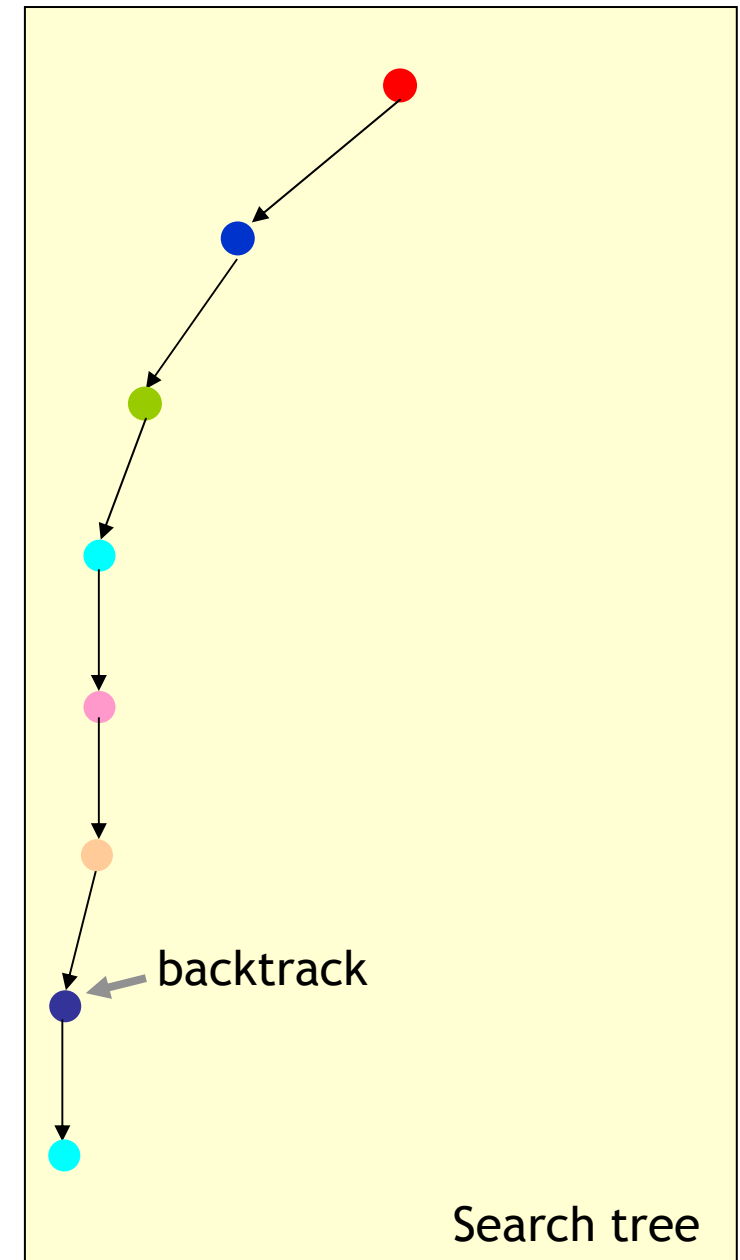
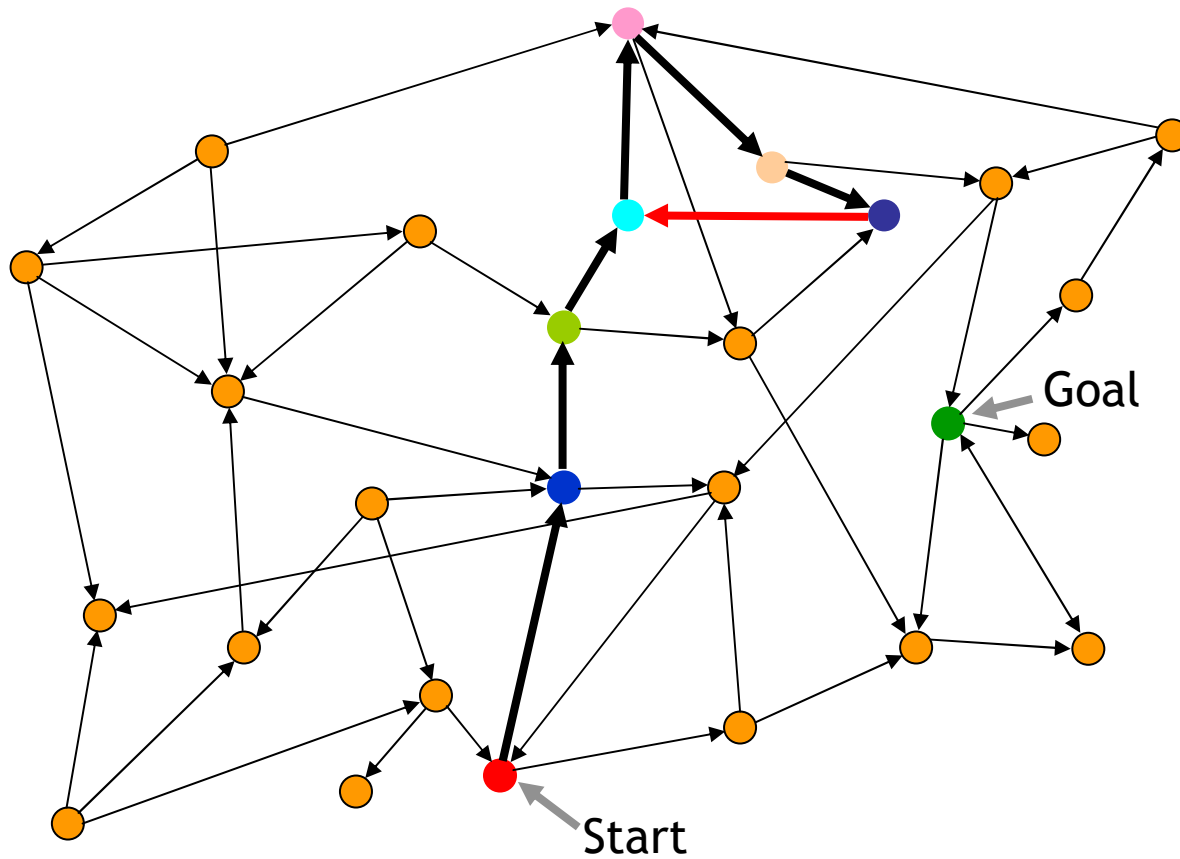
Ricerca nello spazio degli stati



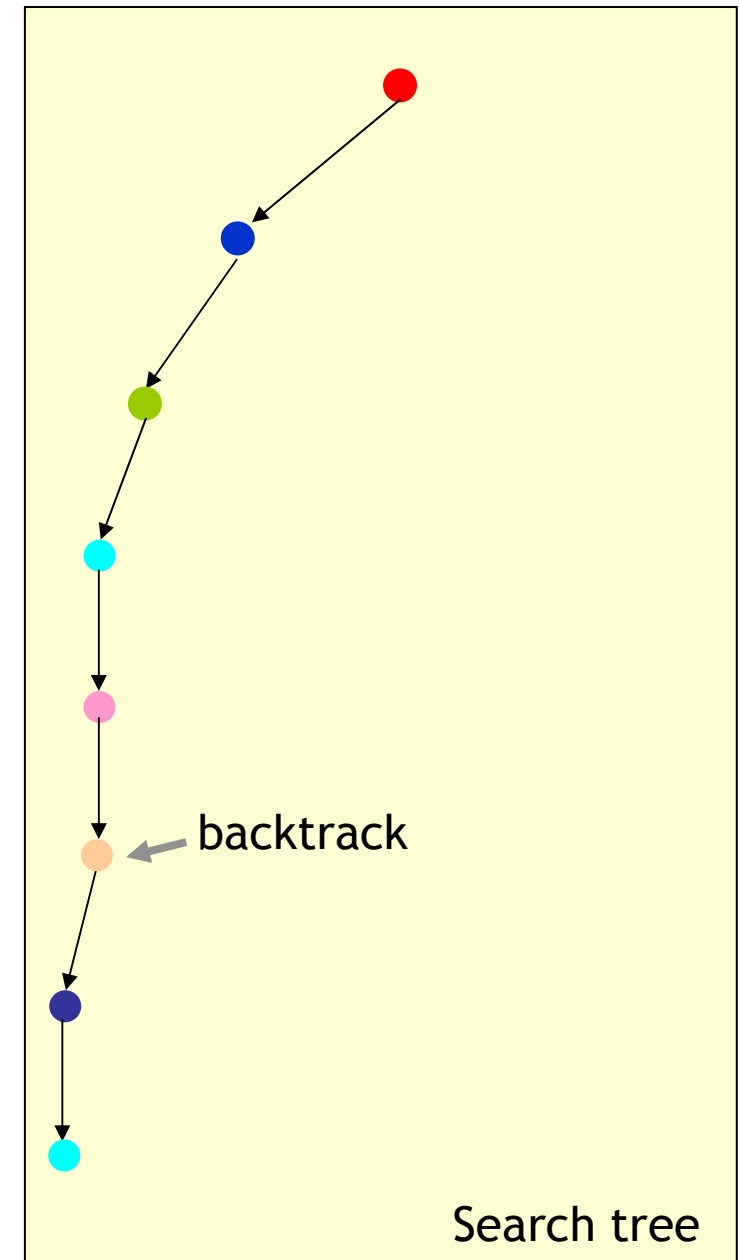
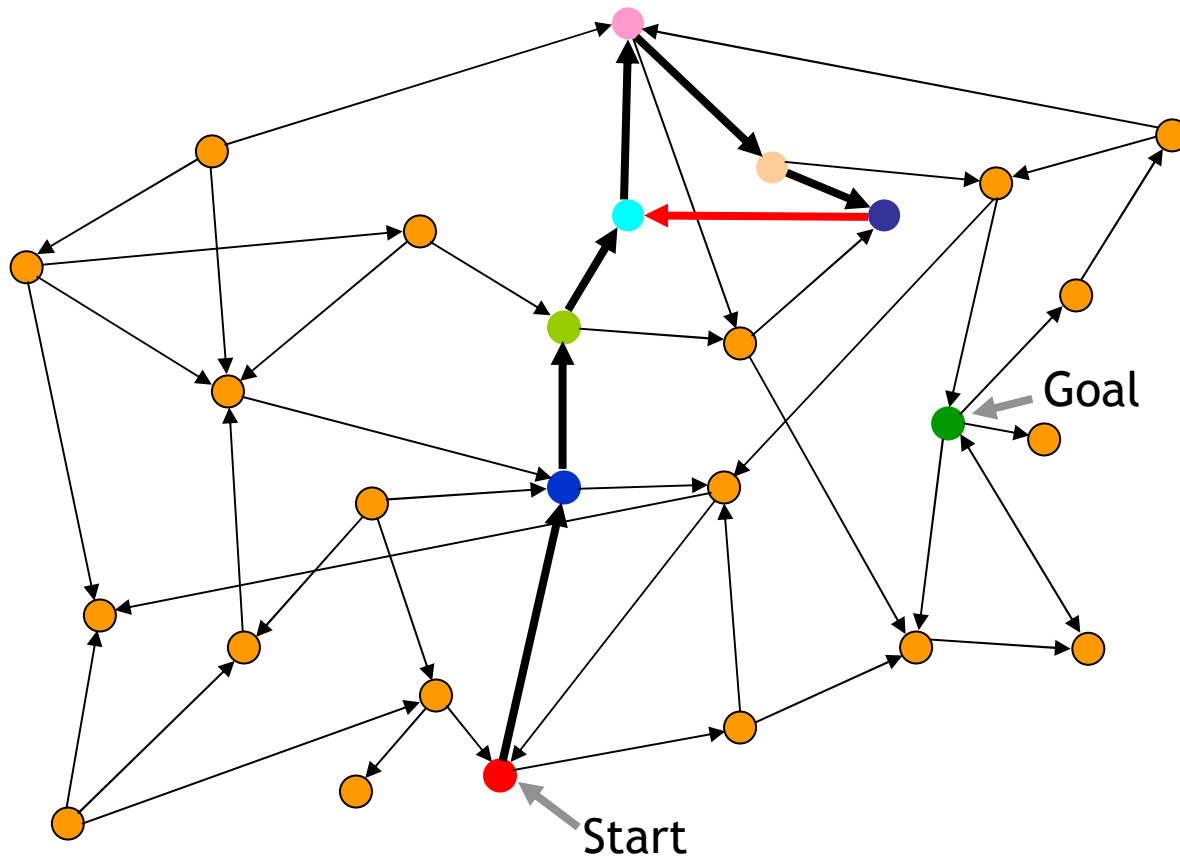
Ricerca nello spazio degli stati



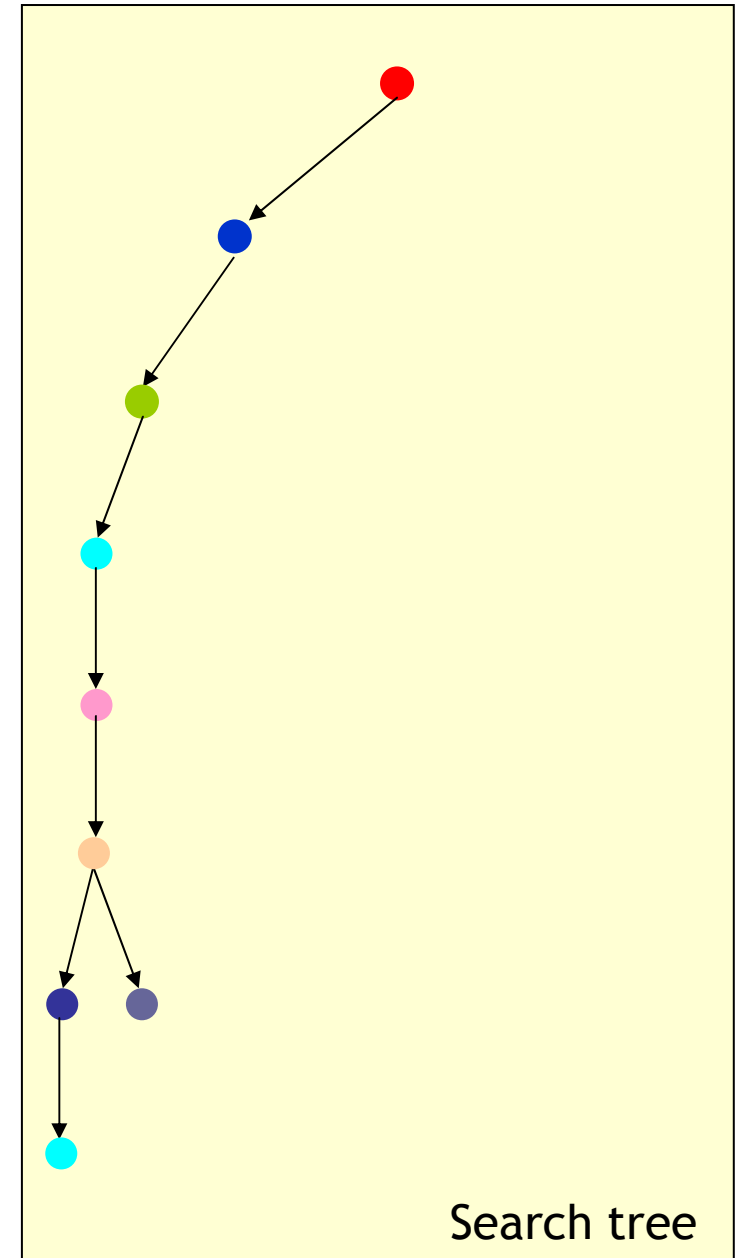
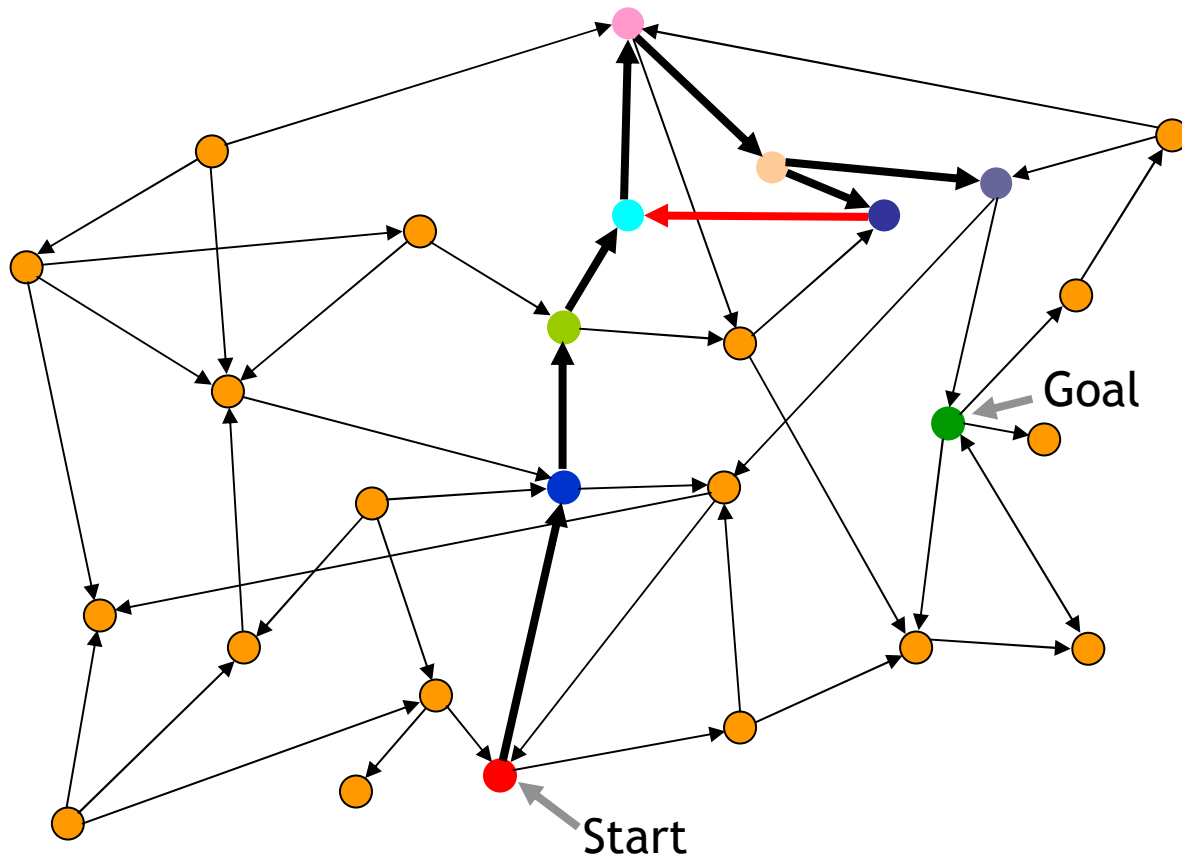
Ricerca nello spazio degli stati



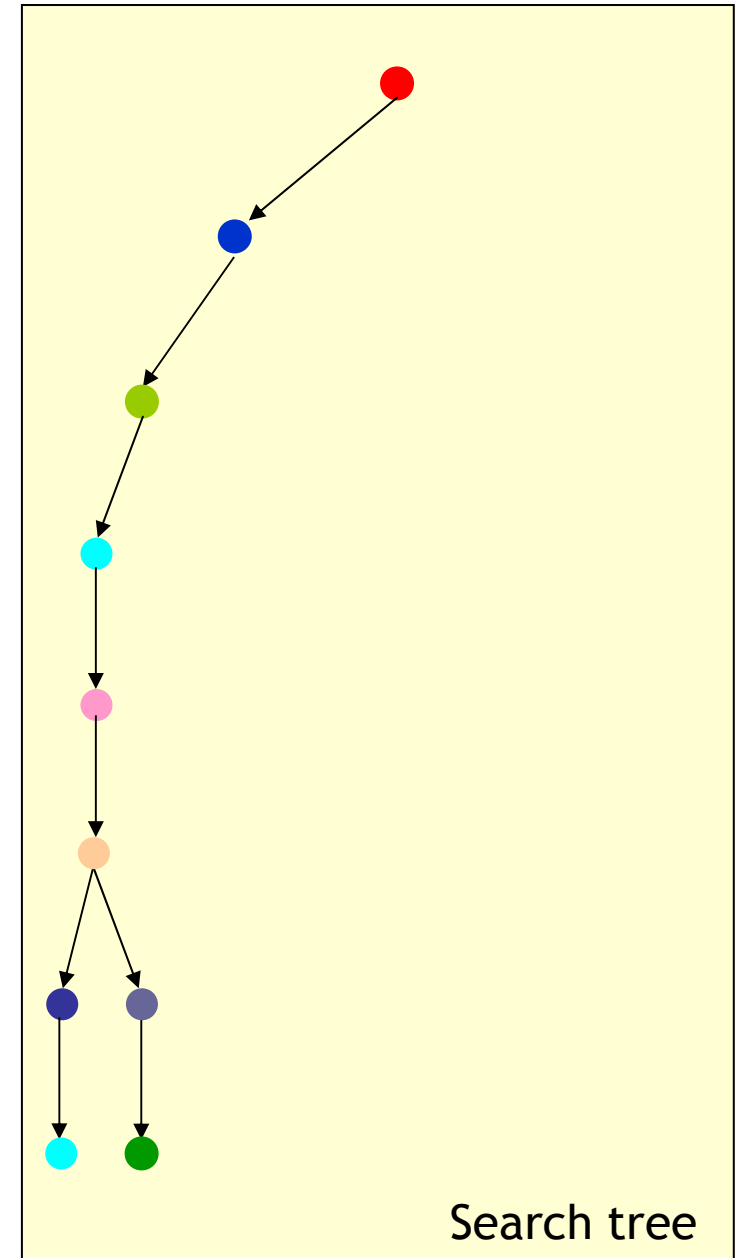
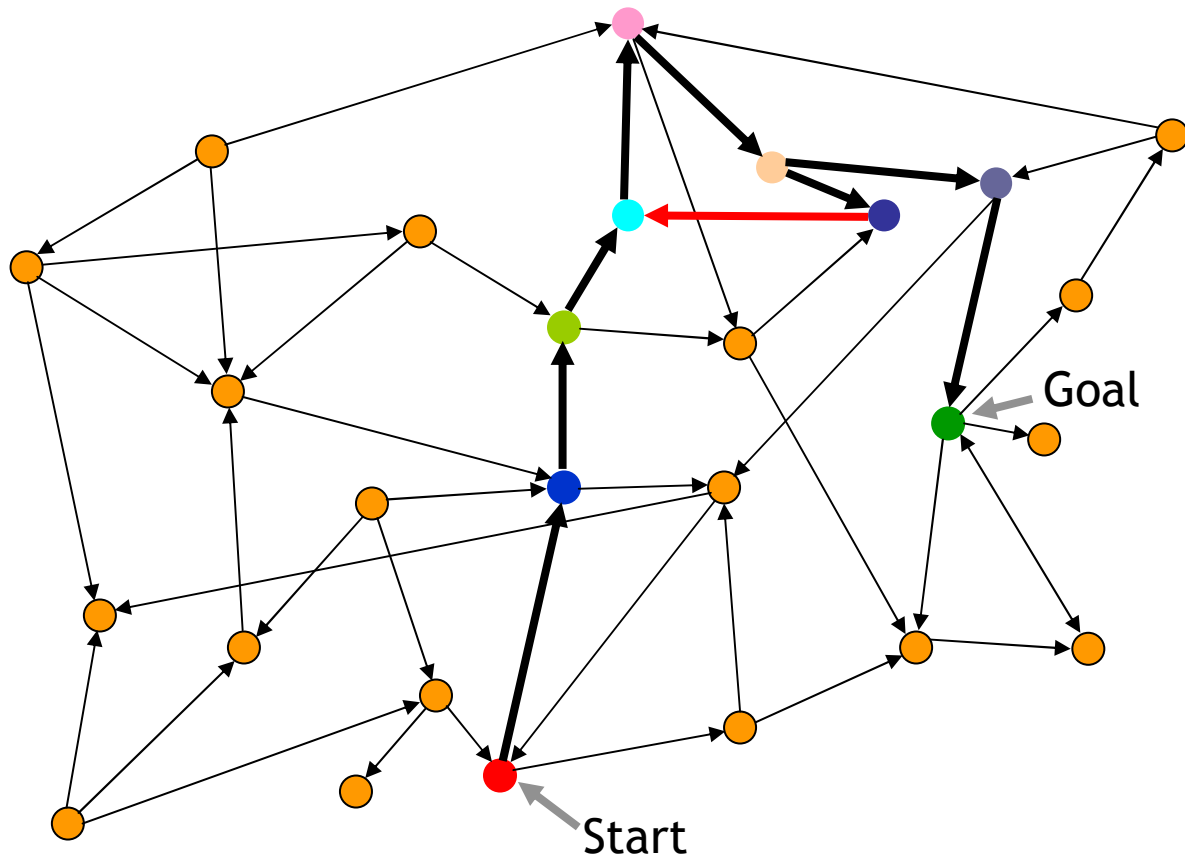
Ricerca nello spazio degli stati



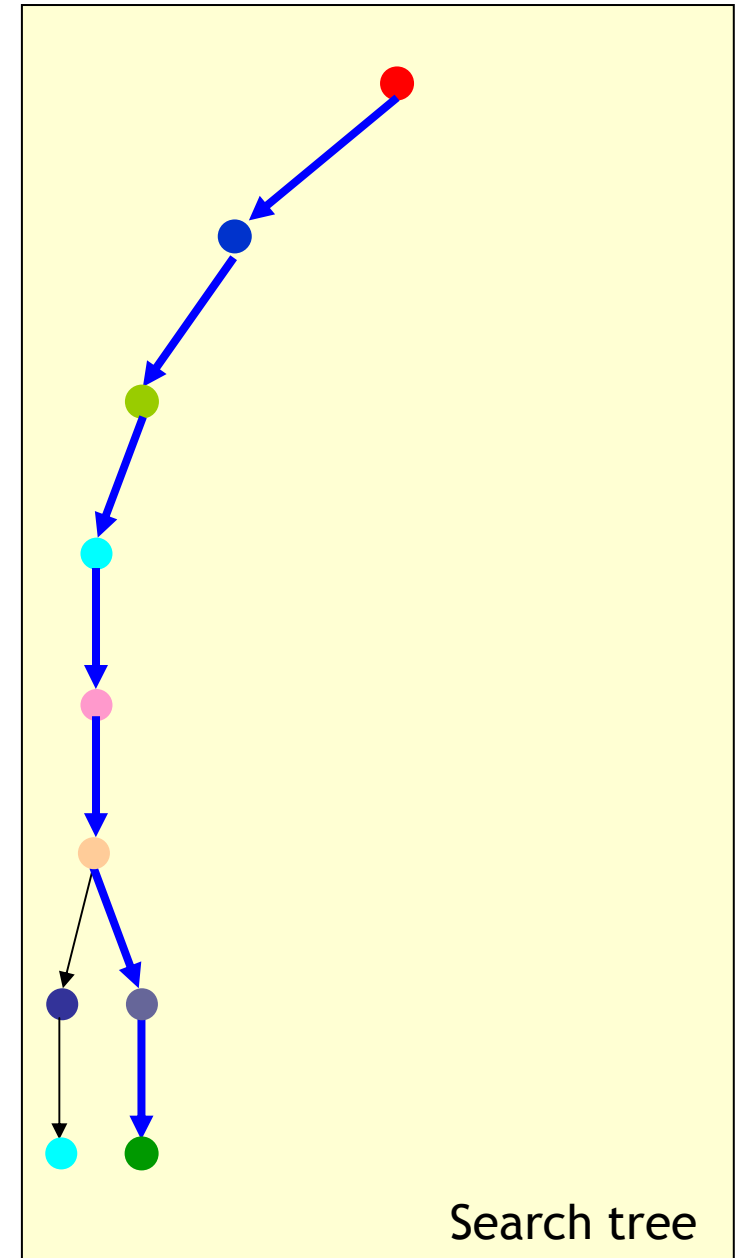
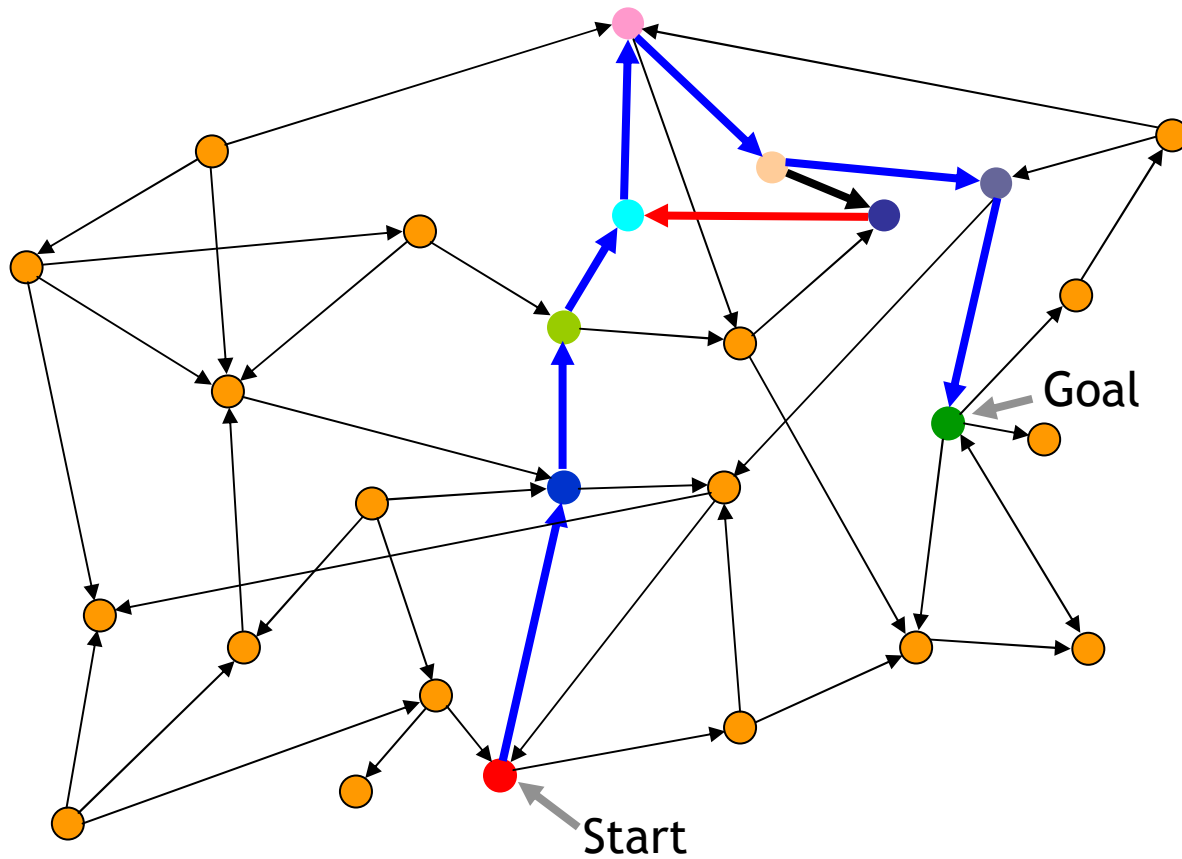
Ricerca nello spazio degli stati



Ricerca nello spazio degli stati



Ricerca nello spazio degli stati



Search tree

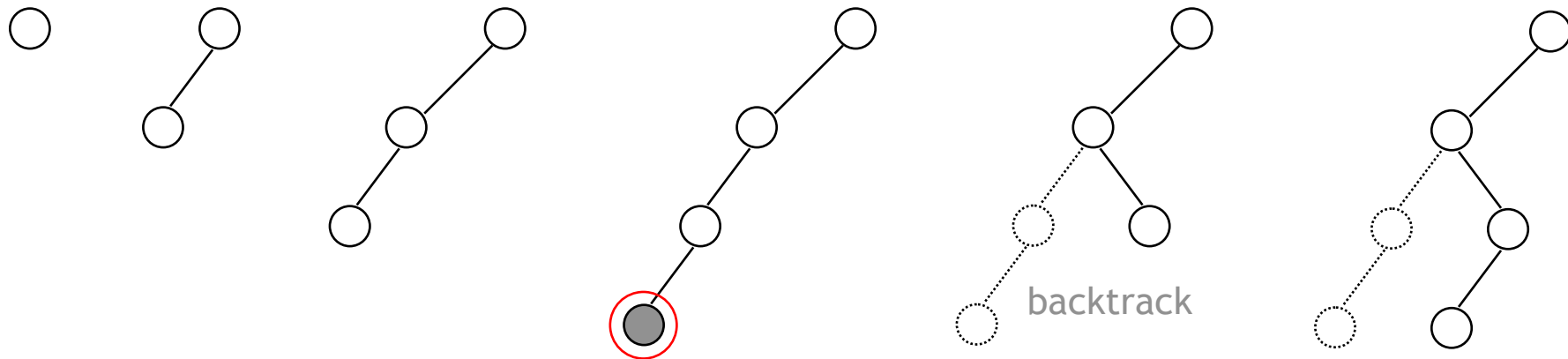
Strategie di ricerca: *depth first*

- Ricerca mirata in profondità

Si espandono i nodi privilegiando la profondità

Si ritrattano (*backtrack*) i rami che non hanno successo

Per la scelta dei nodi da espandere si usa una strategia FIFO (una pila o *stack*)



Completa (se si esplora tutto l'albero)

Ottimale (se si esplora tutto l'albero)

Complessità di tempo: $O(b^d)$

Complessità di memoria: $O(d)$ (se si liberano risorse dopo il backtrack)

Uso: vantaggiosa se si cerca una soluzione

Identificazione dei cicli (*loop checking*)

- Come si identificano i cicli?

Nei due casi:
breadth first
depth first

