

## 13 - Tecniche algoritmiche spazio di ricerca

Per comprendere come progettare un algoritmo, è necessario prima distinguere tre livelli di astrazione:

- **La specifica del problema:** rappresenta la definizione formale della struttura del problema. È importante notare che la specifica definisce il problema in sé, indipendentemente dal metodo o dalla strategia che verrà poi utilizzata per risolverlo.
- **Lo spazio delle soluzioni:** questo concetto è una caratteristica generale del problema. Sebbene contenga idealmente tutte le possibili risposte, lo spazio delle soluzioni è un concetto astratto che non fornisce indicazioni operative su come trovare la soluzione per un caso specifico (una generica istanza).
- **Lo spazio di ricerca:** A differenza dello spazio delle soluzioni, lo spazio di ricerca è lo strumento operativo vero e proprio. Esso ci permette di "caratterizzare" le potenziali soluzioni relative a una generica **istanza** del problema. La sua funzione principale è quella di orientare la scelta dell'algoritmo risolutivo. In altri termini, è il ponte tra il problema astratto e la sua risoluzione pratica.

### Formalizzazione dello spazio di ricerca

Lo spazio di ricerca è definito come strumento concettuale indispensabile per analizzare le potenziali soluzioni di una generica istanza, con il fine ultimo di poter formulare un algoritmo risolutivo.

Definire lo spazio di ricerca un problema  $P$  significa stabilire un metodo preciso che, per ogni specifica istanza  $i$  del problema, definisce un insieme a cui associate due funzioni fondamentali:

1. **Funzione di ammissibilità:** questa funzione ha il compito di verifica. Essa controlla se un elemento preso dallo spazio di ricerca corrisponde effettivamente a una soluzione valida per l'istanza  $i$  che stiamo esaminando.
2. **Funzione di risposta:** questa funzione si occupa dell'estrazione del risultato. Essa ottiene partendo dagli elementi presenti nello spazio di ricerca, le risposte corrispondenti per l'istanza  $i$ .

### Definizione:

Consideriamo un'istanza  $i$  di un problema  $P$ , definito dalla quintupla

$P = \langle I, S, R, S \cup \{\perp\}, Q \rangle$  (dove  $I$  sono gli input,  $S$  le soluzioni,  $R$  la relazione tra input e soluzioni,  $\perp$  indica nessuna soluzione e  $Q$  è il quesito).

Uno spazio di ricerca per l'istanza  $i$  del problema  $P$  è costituito da tre elementi fondamentali e deve soddisfare due condizioni specifiche.

## Elementi costitutivi:

Uno spazio di ricerca è composto da:

**Un insieme  $Z_i$ :** È l'insieme di base che contiene tutti i candidati (gli elementi) che esploreremo. A questo insieme sono associate due funzioni essenziali.

- **La funzione di ammissibilità ( $a$ ):**
  - Notazione:  $a : Z_i \rightarrow \{TRUE, FALSE\}$ : è una funzione booleana che prende in input un elemento  $z$  dello spazio di ricerca ( $Z_i$ ) e restituisce vero o falso.
- **La funzione di risposta ( $o$ ):**
  - Notazione:  $o : Z_i \rightarrow S$ : è una funzione che mappa un elemento  $z$  dello spazio di ricerca in un elemento dello spazio delle soluzioni  $S$ . Serve a tradurre l'elemento interno dell'algoritmo nella risposta formale del problema.

## Le condizioni di validità

Affinchè  $Z_i$  sia un valido spazio di ricerca per l'istanza  $i$ , le funzioni  $a$  e  $o$  devono rispettare due condizioni logiche:

1. **Corrispondenza tra ammissibilità e soluzione:** per ogni elemento  $z$  appartenente a  $Z_i$ , la funzione di ammissibilità  $a(z)$  restituisce true se e solo se la risposta corrispondente  $o(z)$  è effettivamente una soluzione del problema  $P$  per l'istanza  $i$ . Non possono esserci "falsi positivi" (elementi ammissibili che non sono soluzioni) né "falsi negativi" (soluzioni reali che risultano non ammissibili).
2. **Completezza rispetto all'Esistenza:** l'istanza  $i$  ammette una risposta positiva se e solo se esiste almeno un elemento  $z$  nello spazio di ricerca  $Z_i$  tale che sia ammissibile ( $a(z) = TRUE$ ) e la sua traduzione  $O(z)$  sia una risposta valida a  $P$  per  $I$ . Se il problema ha una soluzione, lo spazio di ricerca deve contenerne traccia. Se lo spazio di ricerca non contiene elementi ammissibili, allora il problema non ha soluzione.

## Rappresentazione computazionale

- Metodo di rappresentazione: non basta definire l'insieme  $Z_i$ . E' necessario definire un metodo concreto per rappresentare ogni elemento di  $Z_i$  mediante una struttura dati (vettore, matrice, albero, ecc.).
- Le funzioni di ammissibilità e di risposta devono essere espresse e calcolabili proprio in termini di questa struttura dati scelta. Questo passaggio è ciò che rende l'algoritmo implementabile su un calcolatore.

## Struttura dello spazio di ricerca per problemi di ricerca

Problemi in cui l'obiettivo è trovare una soluzione valida qualsiasi, senza necessariamente cercare la migliore (a differenza dei problemi di ottimizzazione).

Per capire come costruire un algoritmo, dobbiamo confrontare due insiemi di soluzioni:

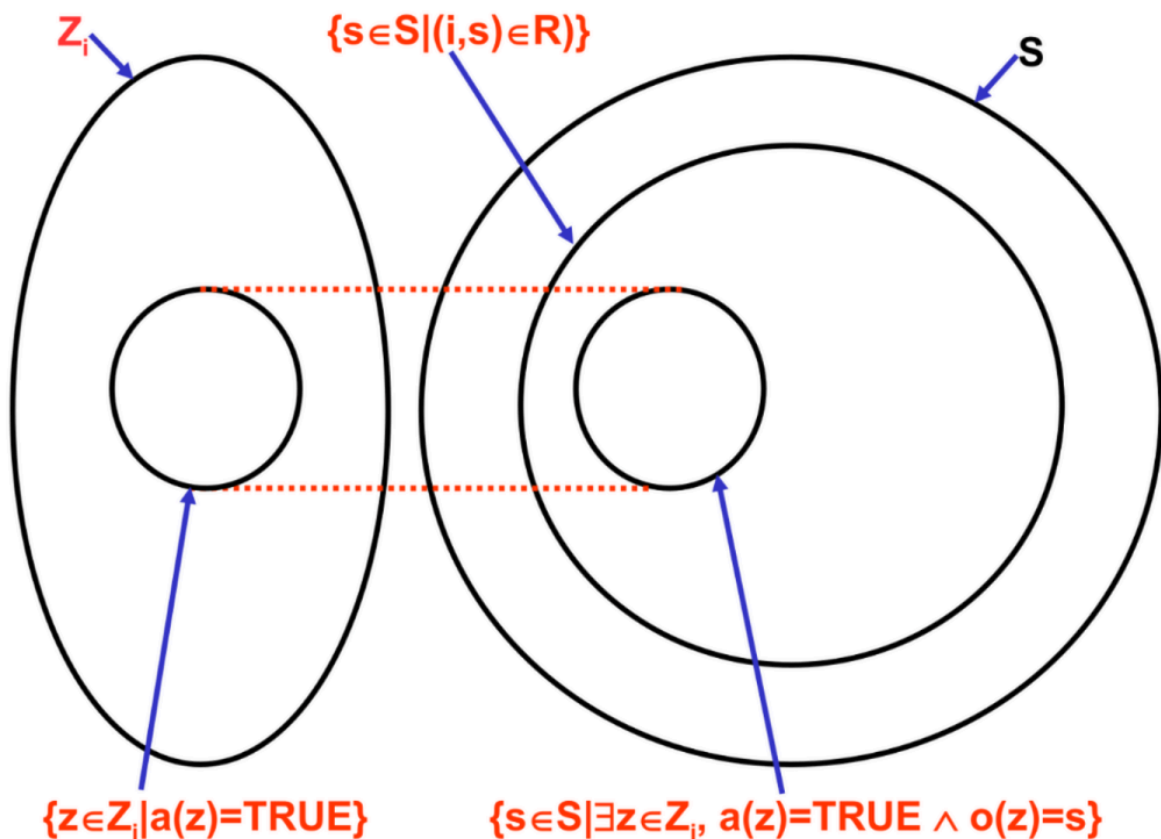
- **L'insieme delle soluzioni reali:** è definito matematicamente come  $\{S \in S \mid (i, s) \in R\}$ . Questo insieme rappresenta tutte le soluzioni teoricamente possibili per l'istanza  $i$ , basandosi esclusivamente sulla specifica del problema  $P$  (la relazione  $R$ ). In un problema di ricerca, questo insieme coincide con tutte le risposte corrette possibili.
- **L'insieme delle soluzioni rappresentate:** è definito come  $\{s \in S \mid \exists z \in Z_i, a(z) = TRUE \wedge o(z) = s\}$ . Questo insieme è costruito dalle soluzioni che il nostro spazio di ricerca  $Z_i$  è effettivamente in grado di "vedere" e generare. Sono quelle soluzioni che possiamo raggiungere attraverso gli elementi  $z$  inclusi nella nostra struttura dati.

**Lo spazio di ricerca non deve necessariamente rappresentare tutte le soluzioni dell'istanza.**

Non è richiesto che l'insieme delle soluzioni rappresentate coincida perfettamente con l'insieme delle soluzioni reali. Costruire uno spazio di ricerca che includa ogni singola soluzione possibile potrebbe essere inutile e dispendioso in termini di risorse.

Poichè stiamo trattando problemi legati alla ricerca, la condizione minima affinché lo spazio di ricerca sia valido è:

- Se esistono risposte positive per l'istanza  $i$  è sufficiente che almeno una di queste soluzioni sia rappresentata nello spazio di ricerca. (Se il problema ha soluzione, il nostro spazio di ricerca deve contenerne almeno una raggiungibile; se ne ignoriamo altre valide, l'algoritmo funzionerà comunque correttamente trovando quella inclusa).



Descrizione dell'esempio:

#### Lato sinistro spazio di ricerca:

L'ovale rappresenta l'intero insieme  $Z_i$  tutti gli elementi candidati che la nostra struttura dati può generare (sia validi che non validi).

All'interno dell'ovale troviamo un cerchio, questo rappresenta l'insieme:

$$\{z \in Z_i \mid a(z) = \text{TRUE}\}$$

Questi sono gli elementi specifici dello spazio di ricerca che la funzione di ammissibilità giudica corretti.

#### Lato destro spazio delle soluzioni:

L'ovale grande a destra rappresenta l'insieme  $S$ , ovvero l'insieme di tutte le possibili risposte previste dal problema.

All'interno di  $S$ , vediamo due sottoinsiemi concentrici ne distinguiamo:

- L'insieme delle soluzioni reali: indicato dalla formula:  $\{s \in S \mid (i, s) \in R\}$ , questo insieme racchiude tutte le soluzioni vere ed esistenti per l'istanza  $i$ .
- L'insieme delle soluzioni rappresentate: indicato dalla formula  $\{s \in S \mid \exists z \in Z_i, a(z) = \text{TRUE} \wedge o(z) = s\}$ , questo insieme contiene le soluzioni che i

nostro algoritmo è effettivamente in grado di trovare. Sono le risposte  $s$  che derivano dagli elementi ammissibili in  $Z_i$  tramite la funzione di risposta  $o(z)$ .

## Problemi di ottimizzazione

I **Problemi di ottimizzazione**, in questo contesto, non cerchiamo solo una soluzione corretta ma la migliore possibile (ad esempio il percorso più breve o il costo minimo).

Anche nel problema di ottimizzazione, la definizione formale di ammissibilità rimane legata alla validità della soluzione, non alla sua qualità.

La funzione di ammissibilità deve assicurare che:

$$\forall z \in Z_i, a(z) = TRUE, \iff (i, o(z)) \in R$$

Un elemento  $z$  è considerato ammissibile se e solo se la corrispondente risposta  $o(z)$  soddisfa la relazione caratteristica  $R$  del problema.

Pochè l'obiettivo di un problema di ottimizzazione è calcolare una soluzione ottimale, cambiano i requisiti minimi che lo spazio di ricerca deve soddisfare.

Non è sufficiente che lo spazio di ricerca contenga soluzioni valide, è necessario che esso includa almeno un elemento la quale si possa ottenere una soluzione ottimale per l'istanza considerata.

## Struttura Formale dello Spazio di Ricerca per l'Ottimizzazione

Per un problema di ottimizzazione, non basta sapere quali soluzioni sono valide; dobbiamo identificare quali sono "ottimali" (le migliori) e garantire che il nostro algoritmo possa trovarne almeno una.

Per un problema di ottimizzazione:

$$\{ s \in S \mid (i, s) \in R \wedge \neg \exists s' \in S ((i, s') \in R \wedge m(i, s') \subseteq m(i, s)) \}$$

Spiegazione:

- $(i, s) \in R$ : la soluzione  $s$  deve essere innanzitutto una soluzione valida per l'istanza  $i$ .
- $\neg \exists s' \in S ((i, s') \in R \wedge m(i, s') \subseteq m(i, s))$  qui  $m$  rappresenta la funzione di misura che valuta la "bontà" della soluzione. La relazione indica che  $s'$  sarebbe "migliore" di  $s$ . La formula afferma che non esiste nessuna soluzione  $s'$  che sia migliore di  $s$ .
- Questo insieme contiene tutte e sole le risposte che massimizzano o minimizzano la funzione obiettivo richiesta dal problema.

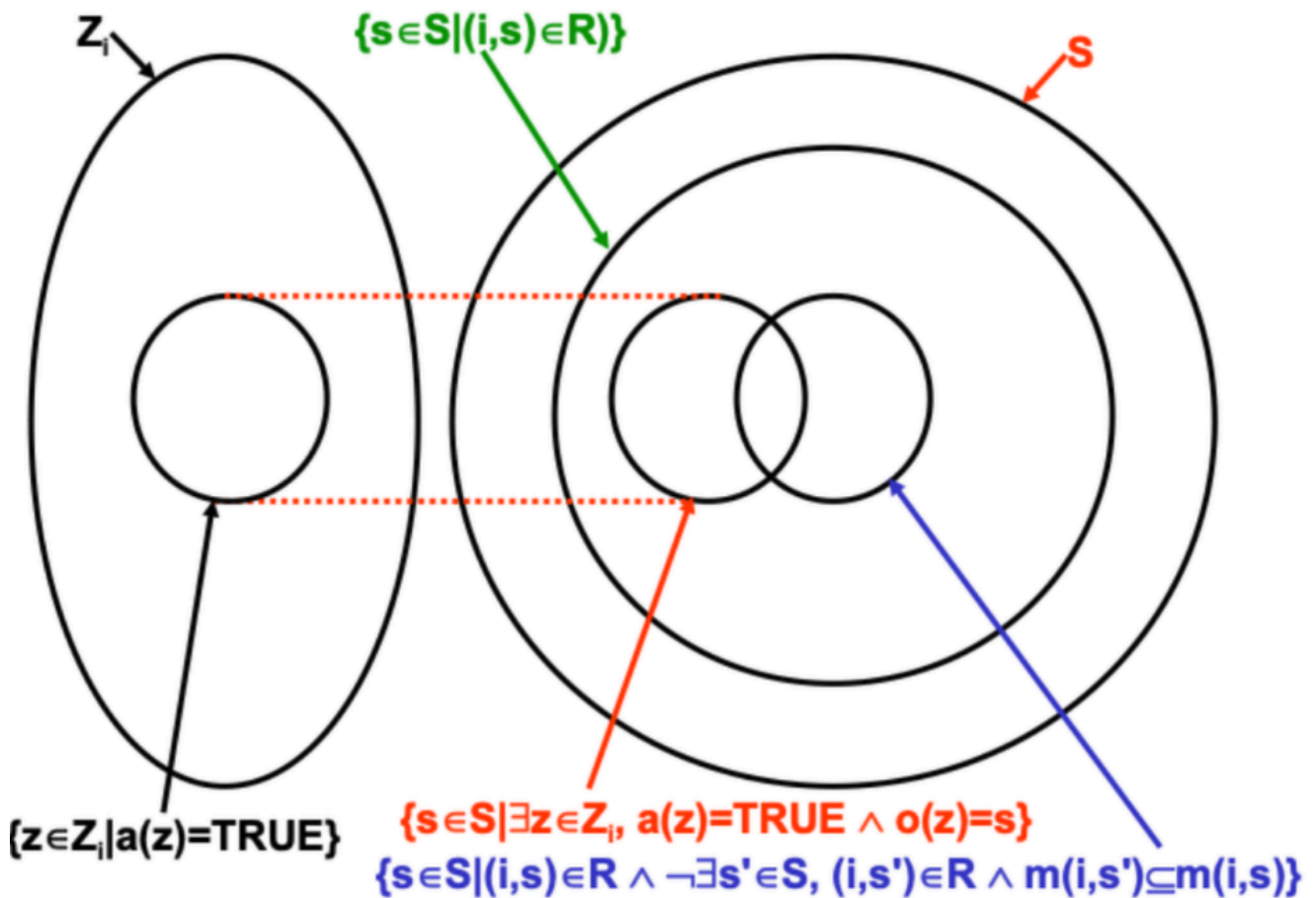
Affinchè uno spazio di ricerca  $Z_i$  sia valido per un problema di ottimizzazione, deve rispettare una condizione rigorosa:

- Deve contenere la rappresentazione di almeno una tra le soluzioni ottimali esistenti.

Se lo spazio di ricerca escludesse tutte le soluzioni ottimali, l'algoritmo fallirebbe nel suo scopo principale: trovare l'ottimo.

In termini insiemistici, la condizione di validità si traduce nel fatto che l'intersezione tra due insiemi non deve essere vuota:

- L'insieme delle soluzioni ottimali reali.  
 $\{s \in S \mid (i, s) \in R \wedge \neg \exists s' \in S ((i, s') \in R \wedge m(i, s') \subseteq m(i, s))\}$
- L'insieme delle soluzioni rappresentate nello spazio di ricerca:  
 $\{s \in S \mid \exists z \in Z_i, a(z) = TRUE \wedge o(z) = s\}$



## Esempio ricerca di un elemento in un vettore

**Listanza del problema:** consideriamo una generica istanza costituita da due input fondamentali:

- Un vettore  $V$  contenente  $n$  elementi.
- Un numero  $m$  (il valore che stiamo cercando).

**Identificazione dello spazio di ricerca ( $Z_i$ ):** Lo spazio di ricerca associato a tale istanza è costituito dall'insieme dei valori  $j$  che rappresenta gli indici del vettore (ovvero gli interi compresi tra 1 e  $n$ ). In questo contesto, l'obiettivo non è individuare il valore  $m$ , bensì determinare la posizione all'interno di una struttura dati.

**Definizione delle funzioni e della struttura dati:** Una volta identificato l'insieme, si definiscono gli strumenti operativi per l'esplorazione:

- **Funzione di ammissibilità:** Dato un generico indice  $j$ , la funzione verifica la condizione  $V[j] = m$ . Essa controlla, se l'elemento del vettore alla posizione  $j$  coincide con il valore creato.
- **Funzione di risposta:** nel caso in cui la verifica abbia esito positivo, la funzione restituisce il valore  $j$ , che ricostruisce la risposta all'istanza.
- Per rappresentare i singoli elementi di questo insieme è sufficiente utilizzare una semplice variabile di tipo intero.

## Differenze tra spazio delle soluzioni e spazio di ricerca per un problema P

È fondamentale operare una distinzione formale tra questi due concetti, che agiscono su livelli di astrazione differenti:

- **Spazio delle soluzioni:** esso fa riferimento globalmente a tutte le possibili soluzioni ammissibili per il problema astratto, indipendentemente dalla specifica istanza da risolvere.
- **Spazio di ricerca:** costituisce lo strumento operativo utilizzato per caratterizzare le soluzioni di ogni singola istanza del problema: La sua definizione determina:
  1. Una struttura dati per rappresentare dei candidati.
  2. Un meccanismo di verifica, tramite la formulazione della funzione di ammissibilità in termini di tale struttura (per accertare se una configurazione corrisponde a una soluzione).
  3. Un metodo per derivare la risposta finale tramite la funzione di risposta.

## Criteri per la definizione dello spazio di ricerca

La determinazione di uno spazio di ricerca per un problema P non è univoca: esistono diverse modalità di costruzione. La scelta di una rappresentazione rispetto a un'altra è determinante per l'efficienza dell'algoritmo. I criteri per valutare la qualità di uno spazio di ricerca sono i seguenti:

1. **Non Ridondanza:** Lo spazio di ricerca deve caratterizzare le soluzioni a un'istanza in modo non ridondante. È opportuno selezionare una struttura dati che escluda a priori configurazioni che non possono corrispondere ad alcuna soluzione, riducendo così l'insieme dei candidati da esaminare.
2. **Significatività Strutturale:** Lo spazio di ricerca non deve limitarsi a una banale riformulazione del problema o della sua relazione caratteristica, né deve occultarne la complessità. Esso deve fornire elementi significativi per la comprensione della struttura del problema e delle sue soluzioni

## Esempio delle $n$ regine

L'obiettivo è posizionare  $i$  regine su una scacchiera di dimensione  $i \times i$  in modo tale che nessuna regina possa attaccarne un'altra.

In una prima analisi, si può definire lo spazio di ricerca in modo intuitivo basandosi sulla rappresentazione visiva della scacchiera.

Sia  $i$  la dimensione della scacchiera:

- **Insieme ( $Z_i$ ):** Lo spazio di ricerca è costituito dall'insieme di tutte le possibili configurazioni della scacchiera di dimensione  $i \times i$  in cui sono posizionate  $i$  regine, senza alcuna restrizione iniziale.
- **Funzione di Ammissibilità:** Questa funzione deve effettuare un controllo esaustivo. Verifica se la configurazione soddisfa tutti i vincoli del problema, ovvero accerta che non esistano due regine posizionate sulla stessa riga, sulla stessa colonna o sulla stessa diagonale.
- **Funzione di Risposta:** È la funzione identità; la configurazione ammissibile trovata costituisce essa stessa la soluzione.
- **Struttura Dati:** La rappresentazione più immediata per ogni elemento dello spazio di ricerca è una **matrice**  $i \times i$  di caratteri, dove un simbolo specifico (es. 'q') indica la presenza di una regina e uno spazio vuoto indica una casella libera.

È possibile migliorare la rappresentazione precedente applicando il criterio di "non ridondanza"? La risposta risiede nel cambiare la struttura dati per escludere a priori configurazioni impossibili.

È possibile migliorare la rappresentazione precedente applicando il criterio di "non ridondanza"?

La risposta risiede nel cambiare la struttura dati per escludere a priori configurazioni impossibili.

Poiché le regole degli scacchi impongono che non vi possano essere due regine sulla stessa riga, è superfluo utilizzare una matrice che permetterebbe di posizionarne più di una per riga.

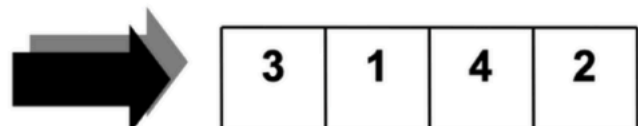
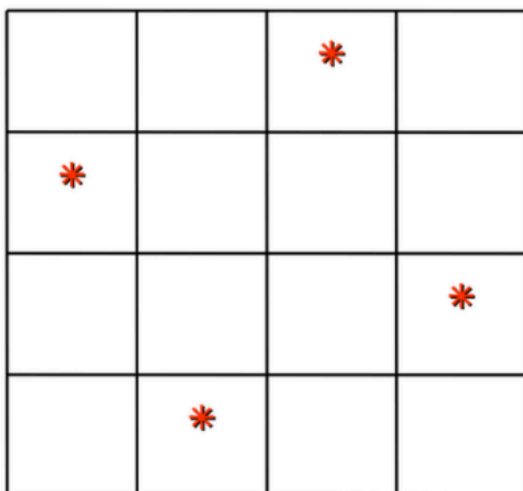
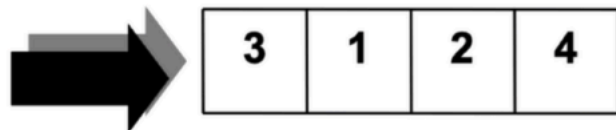
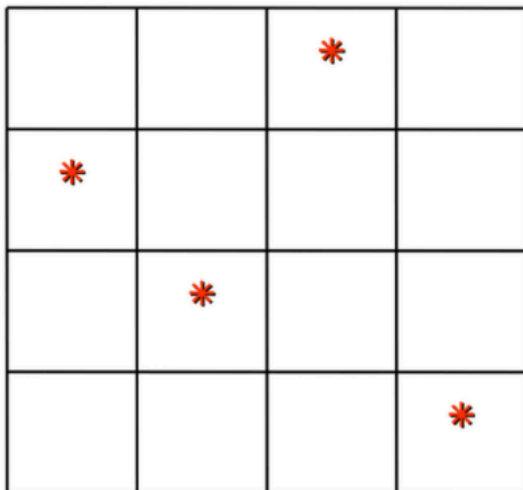
Si adotta quindi una rappresentazione più efficiente:

- **Nuova Struttura Dati:** Si utilizza un vettore  $V$  di dimensione  $i$  con elementi interi compresi tra 1 ed  $i$ .
- **Semantica della Struttura:** L'indice del vettore rappresenta la riga, mentre il valore contenuto rappresenta la colonna.
  - Se l'elemento in posizione  $j$  del vettore assume valore  $h$  (ovvero  $V[j] = h$ ), significa che la regina della riga  $j$  è posizionata nella colonna  $h$ .
- **Vantaggi dell'Ottimizzazione:**
  1. **Vincolo di Riga:** Soddisfatto intrinsecamente dalla struttura dati (ogni posizione  $j$  del vettore può contenere un solo valore).

2. **Vincolo di Colonna:** Per escludere configurazioni con due regine sulla stessa colonna, è sufficiente limitare lo spazio di ricerca ai vettori che non presentano valori ripetuti (ovvero le permutazioni degli interi da 1 a  $i$ ).

Questo passaggio dimostra come una scelta oculata dello spazio di ricerca (passando da matrice a vettore) riduca drasticamente il numero di controlli necessari nella funzione di ammissibilità, che ora dovrà preoccuparsi principalmente delle diagonalità.

## Ad esempio:



c

Allora, per il problema delle  $n$  regine:

Sfruttando la rappresentazione vettoriale ottimizzata, è possibile definire formalmente le componenti dello spazio di ricerca come segue:

- L'insieme che costituisce lo spazio di ricerca coincide con tutti i possibili modi di memorizzare i valori interi da 1 a  $i$  in un vettore  $V$  di dimensione  $i$ . In termini matematici,

questo insieme corrisponde a tutte le possibili **permutazioni** della sequenza  $(1, \dots, i)$ . Ogni elemento dello spazio è dunque un vettore in cui l'indice rappresenta la riga e il valore contenuto rappresenta la colonna, garantendo implicitamente che non vi siano conflitti di riga o colonna.

- **La Funzione di Ammissibilità:** Dato un generico elemento dello spazio di ricerca (il vettore  $V$ ), la funzione deve verificare il rispetto dei vincoli sulle diagonali. La configurazione è ammissibile se il vettore  $V$  **non contiene** due elementi in posizioni distinte  $h$  e  $k$  tali che si verifichi la seguente condizione:

$$|V[h] - V[k]| = |h - k|$$

Questa equazione identifica geometricamente due regine poste sulla stessa diagonale. La funzione di ammissibilità restituisce *TRUE* solo se nessuna coppia di indici soddisfa tale uguaglianza.

- **La Funzione di Risposta:** Questa funzione ha un compito di ricostruzione: a partire da un generico vettore  $V$  validato, essa genera la corrispondente configurazione completa della scacchiera, traducendo la rappresentazione compatta nella soluzione finale del problema.

## Paradigma generativo

Le tecniche di progettazione degli algoritmi possono essere categorizzate in base al **paradigma di utilizzo dello spazio di ricerca**.

Esistono quattro tecniche principali:

- La tecnica di enumerazione.
- La tecnica di backtracking.
- La tecnica greedy.
- La tecnica divide et impera.

Queste tecniche vengono raggruppate in due macro-paradigmi fondamentali, distinti dalla modalità con cui interagiscono con l'insieme dei candidati alla soluzione:

1. Il **Paradigma Selettivo**.
2. Il **Paradigma Generativo**.

### Paradigma Selettivo

Questo paradigma comprende tecniche che visitano lo spazio di ricerca tentando di individuare un elemento ammissibile per l'istanza considerata.

- L'algoritmo fa riferimento all'intero spazio di ricerca, il quale viene esplorato con sistematicità secondo una modalità definita.
- Appartengono a questo paradigma la tecnica enumerativa e la tecnica di backtracking.

### Paradigma Generativo

Questo paradigma raggruppa tecniche che generano direttamente la soluzione senza selezionarla tra gli elementi dello spazio di ricerca .

- **Ruolo dello Spazio di Ricerca:** Esso è considerato esclusivamente in fase di progetto dell'algoritmo, allo scopo di caratterizzare le soluzioni e definire una strategia risolutiva diretta per ogni istanza.
- Appartengono a questo paradigma la tecnica greedy e la tecnica divide et impera.