Corso di Laurea in INFORMATICA

Algoritmi e Strutture Dati a.a. 2012-2013

MODULO 14

TECNICHE ALGORITMICHE 2

Paradigma selettivo: la tecnica enumerativa e la tecnica di backtracking

LE TECNICHE PER PROGETTARE ALGORITMI SI POSSONO CARATTERIZZARE IN BASE AL MODO NEL QUALE UTILIZZANO LO SPAZIO DI RICERCA.

DAL PARADIGMA *SELETTIVO* SONO CREATE TECNICHE DI PROGETTO DI ALGORITMI CHE, PER L'ISTANZA DEL PROBLEMA PRESA IN CONSIDERAZIONE, VISITANO LO SPAZIO DI RICERCA TENTANDO DI TROVARE UN ELEMENTO AMMISSIBILE.

OGNI ALGORITMO FA RIFERIMENTO ALL'INTERO SPAZIO DI RICERCA CHE VIENE ESPLORATO CON SISTEMATICITÀ IN UNA DEFINITA MODALITÀ.

APPARTENGONO A QUESTO PARADIGMA LA TECNICA ENUMERATIVA E QUELLA DI BACKTRACKING.

ESEMPIO

ORDINAMENTO DI UN VETTORE DI INTERI SECONDO L’ORDINE NON DECRESCENTE DEI SUOI ELEMENTI

IL VETTORE DA ORDINARE È V.

LO SPAZIO DI RICERCA È COSTITUITO DALLE PERMUTAZIONI DI V.

LA FUNZIONE DI AMMISSIBILITA’ VERIFICA CHE NON ESISTANO PERMUTAZIONI э’ SE h > k, SIA

V(h) < V(K)

LA FUNZIONE DI RISPOSTA È L’IDENTITÀ. SE N È LA DIMENSIONE DI V LO SPAZIO DI RICERCA HA DIMENSIONE N !

RICORDIAMO CHE ***UN ALGORITMO SELETTIVO*** EFFETTUA LA VISITA DELLO SPAZIO DI RICERCA PER TROVARE UNA PERMUTAZIONE CHE SODDISFI LA CONDIZIONE DI AMMISSIBILITÀ MENTRE ***UN ALGORITMO GENERATIVO*** DERIVA LA SOLUZIONE CON UN PROCEDIMENTO DIRETTO SULLA ISTANZA, SENZA VISITARE LO SPAZIO DI RICERCA.

SIA IL VETTORE V(1), V(2), V(3)

SONO POSSIBILI 6 ORDINAMENTI

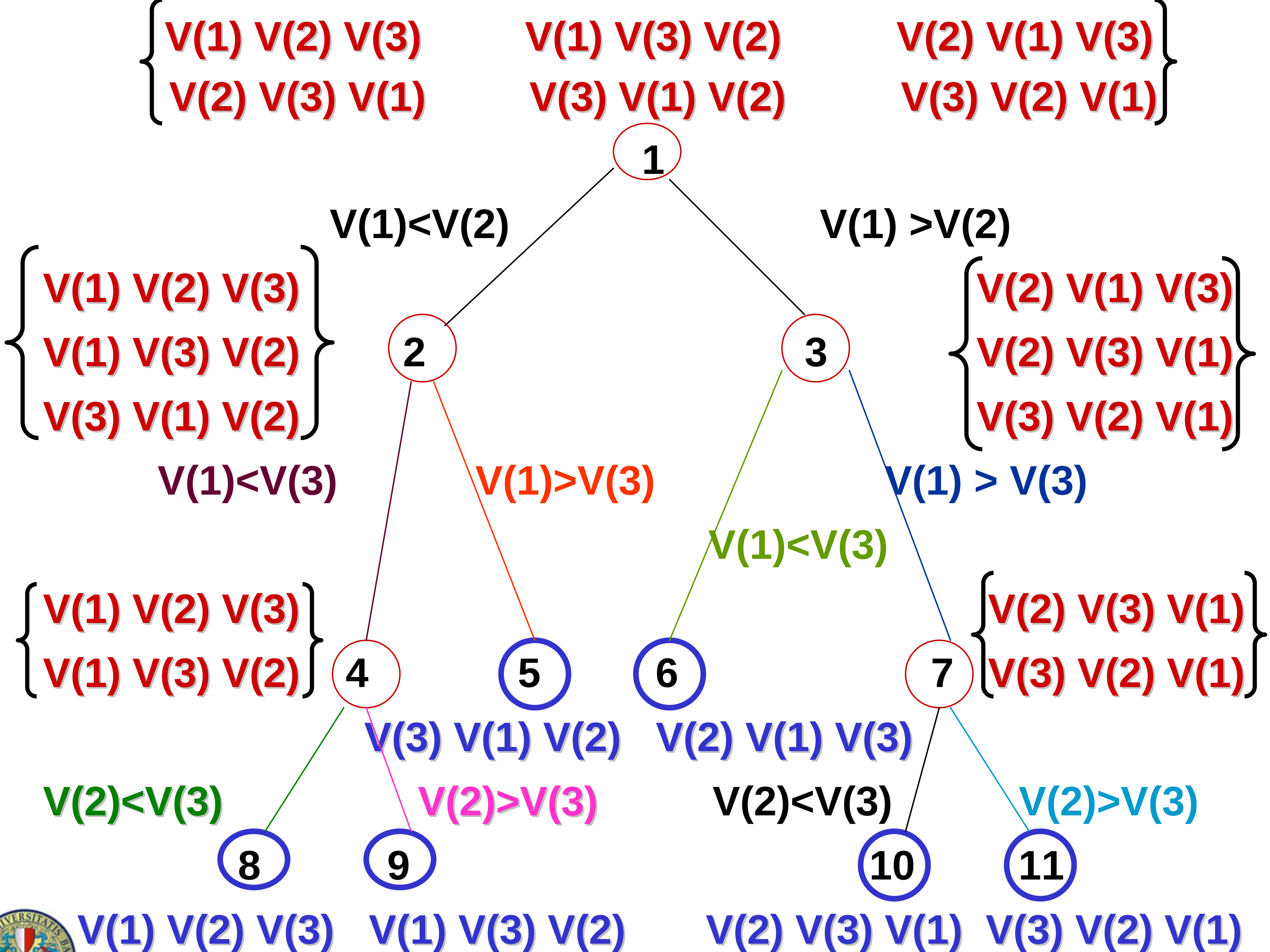
V(1) V(2) V(3)	V(1) V(3) V(2)	V(2) V(1) V(3)
V(2) V(3) V(1)	V(3) V(1) V(2)	V(3) V(2) V(1)

NELL' ESEMPIO LA FUNZIONE DI AMMISSIBILITÀ SUGGERISCE CHE LA SOLUZIONE COINCIDE CON QUELLA PERMUTAZIONE CHE NON HA INVERSIONI.

SE LO SPAZIO DI RICERCA E' RAPPRESENTATO COME UN GRAFO I CUI NODI SONO TUTTE LE POSSIBILI **CONFIGURAZIONI O PERMUTAZIONI DI V** POSSIAMO RAPPRESENTARE IL ***PROCEDIMENTO DI ESPLORAZIONE DELLO SPAZIO DI RICERCA*** MEDIANTE UN ALBERO, CHE HA NEI NODI I CONFRONTI UTILI A SCEGLIERE QUELLE PERMUTAZIONI CHE SONO SOLUZIONI.

LO SPAZIO VIENE PARTIZIONATO E **CIASCUNA FOGLIA DELL'ALBERO RAPPRESENTA UN POSSIBILE ORDINAMENTO.**

LA **PROFONDITÀ** DELL'ALBERO RAPPRESENTA IL NUMERO DI CONFRONTI NECESSARIO NEL CASO PEGGIORE.


V(1) V(2) V(3) V(1) V(3) V(2) V(2) V(1) V(3)
V(2) V(3) V(1) V(3) V(1) V(2) V(3) V(2) V(1)
1
V(1)<V(2)
V(1) >V(2)
V(1) V(2) V(3)
V(1) V(3) V(2)
V(3) V(1) V(2)
2
3
V(2) V(1) V(3)
V(2) V(3) V(1)
V(3) V(2) V(1)
V(1)<V(3)
V(1)>V(3)
V(1)<V(3)
V(1) > V(3)
V(1) V(2) V(3)
V(1) V(3) V(2)
4
5
6
7
V(2) V(3) V(1)
V(3) V(2) V(1)
V(3) V(1) V(2)
V(2) V(1) V(3)
V(2)<V(3)
V(2)>V(3)
V(2)<V(3)
V(2)>V(3)
8
9
10
11
V(1) V(2) V(3)
V(1) V(3) V(2)
V(2) V(3) V(1)
V(3) V(2) V(1)

LA TECNICA ENUMERATIVA

BASATA SULLA SISTEMATICA ISPEZIONE, ELEMENTO PER ELEMENTO, DELLO SPAZIO DI RICERCA ASSOCIATO AD UNA ISTANZA DI UN PROBLEMA, È UNA TECNICA CHE GARANTISCE, SE LO SPAZIO DI RICERCA È FINITO, UNA TERMINAZIONE.

PER UN PROBLEMA DI *RICERCA* SI TERMINA QUANDO SI È INDIVIDUATO UN ELEMENTO AMMISSIBILE O QUANDO È ESAURITO LO SPAZIO DI RICERCA.

PER UN PROBLEMA DI *OTTIMIZZAZIONE* SI DOVRÀ CONFRONTARE, AI FINI DELLA SELEZIONE, TUTTI GLI ELEMENTI AMMISSIBILI E LA TERMINAZIONE È COMUNQUE DATA DALL' ESAURIMENTO DELLO SPAZIO DI RICERCA.

PER GARANTIRE LA POSSIBILITÀ DI UNA VISITA SISTEMATICA DI UNO SPAZIO DI RICERCA SI ASSOCIA AD ESSO UNA RELAZIONE DI ORDINAMENTO TOTALE IN MODO DA DEFINIRE PER LO SPAZIO Z_i ASSOCIATO AD i :

- **UN METODO PER STABILIRE IL PRIMO ELEMENTO DA CONSIDERARE**
- **UN METODO PER STABILIRE L'ELEMENTO SUCCESSIVO**
- **UN METODO PER VERIFICARE SE SI SONO ESAMINATI TUTTI GLI ELEMENTI**

ALGORITMO ENUMERATIVO PER PROBLEMI DI RICERCA

1. CONSIDERA IL PRIMO ELEMENTO x DELLO SPAZIO DI RICERCA
2. SE a(x)=TRUE, ALLORA FORNISCI O(x) COME RISULTATO
3. SE TUTTI GLI ELEMENTI DELLO SPAZIO DI RICERCA SONO STATI ESAMINATI, FORNISCI ⊥ COME RISULTATO
4. ALTRIMENTI CONSIDERA COME NUOVO x L'ELEMENTO SUCCESSIVO DELLO SPAZIO DI RICERCA E RIPETI DAL 2

ESEMPIO: *IL GIOCO DELL'8*

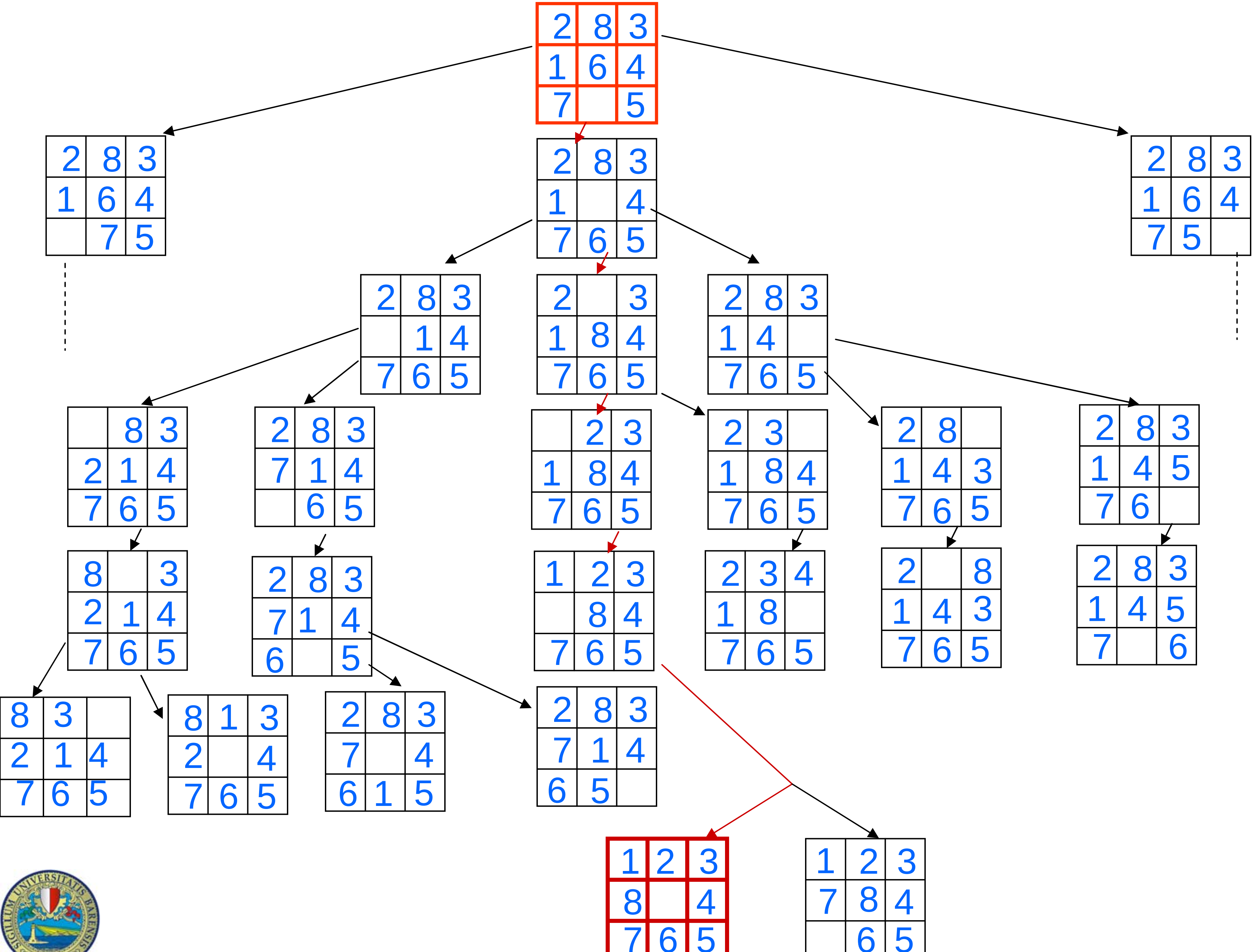
LO STATO INIZIALE

2	8	3
1	6	4
7		5

LO STATO FINALE

1	2	3
8		4
7	6	5

I TASSELLI NUMERATI DEVONO ESSERE DISPOSTI ORDINATI LUNGO I BORDI: LA SOLUZIONE PUO' ESSERE RICERCATA ESAMINANDO <u>TUTTE</u> LE POSSIBILI DISPOSIZIONI GENERATE ATTRAVERSO MOSSE SUCCESSIVE. IL PROCEDIMENTO DI RICERCA SI RAPPRESENTA MEDIANTE UN ALBERO CHE HA NEI NODI LE POSSIBILI CONFIGURAZIONI GENERATE MOSSA DOPO MOSSA.



2 8 3 1 6 4 7 5
2 8 3 1 6 4 7 5
2 8 3 1 4 7 6 5
2 8 3 1 6 4 7 5
2 8 3 1 4 7 6 5
2 3 1 8 4 7 6 5
2 8 3 1 4 7 6 5
8 3 2 1 4 7 6 5
2 8 3 7 1 4 6 5
2 3 1 8 4 7 6 5
2 3 1 8 4 7 6 5
2 8 1 4 3 7 6 5
2 8 3 1 4 5 7 6
8 3 2 1 4 7 6 5
2 8 3 7 1 4 6 5
1 2 3 8 4 7 6 5
2 3 4 1 8 7 6 5
2 8 1 4 3 7 6 5
2 8 3 1 4 5 7 6
8 3 2 1 4 7 6 5
8 1 3 2 4 7 6 5
2 8 3 7 4 6 1 5
2 8 3 7 1 4 6 5
1 2 3 8 4 7 6 5
1 2 3 7 8 4 6 5

ALGORITMO ENUMERATIVO PER PROBLEMI DI OTTIMIZZAZIONE

1. CONSIDERA IL PRIMO ELEMENTO x DELLO SPAZIO DI RICERCA
2. SE a(x)=TRUE, E x È LA PRIMA SOLUZIONE TROVATA, OPPURE SE a(x)=TRUE, E x È MIGLIORE DELLA SOLUZIONE OTTIMA CORRENTE, PONI LA SOLUZIONE OTTIMA CORRENTE =x
3. SE TUTTI GLI ELEMENTI DELLO SPAZIO DI RICERCA SONO STATI CONSIDERATI, ALLORA FORNISCI COME RISULTATO ⊥ SE NON SONO STATE TROVATE SOLUZIONI AMMISSIBILI, OPPURE O(x) SE x È SOLUZIONE OTTIMA CORRENTE
4. ALTRIMENTI CONSIDERA COME NUOVO x L'ELEMENTO SUCCESSIVO DELLO SPAZIO DI RICERCA E RIPETI DAL 2

ESEMPIO

IL GIOCO DELL'OTTO (COME PROBLEMA DI OTTIMIZZAZIONE)

POTREMMO PORCI L'OBIETTIVO NON SOLO DI TROVARE UNA SOLUZIONE QUALSIASI MA DI TROVARNE UNA OTTIMA, PER ESEMPIO, SULLA BASE DEL CRITERIO "IL PRIMA POSSIBILE".

QUESTO SI TRADUCE NELLA POSSIBILITA' DI VALUTARE IL NUMERO DI MOSSE CHE CI PORTA ALLA SOLUZIONE E NELL'ESPLORARE L'INTERO ALBERO ALLA RICERCA DEL *CAMMINO PIU' BREVE* CHE CI PORTA ALLA SOLUZIONE (NUMERO MINIMO DI MOSSE).

LA TECNICA BAKCTRACKING

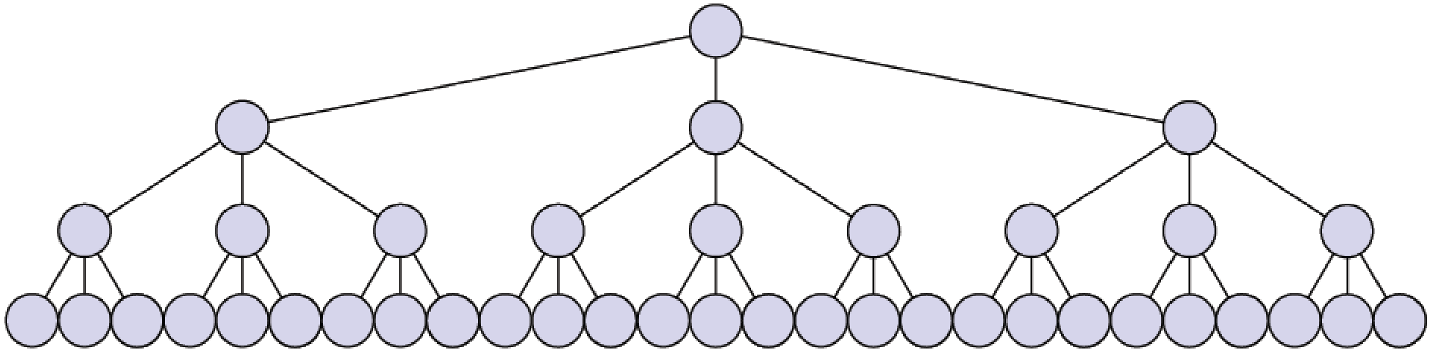
NELLA TECNICA DI BAKCTRACKING LA GENERAZIONE DEGLI ELEMENTI DELLO SPAZIO DI RICERCA DA VISITARE AVVIENE SECONDO UN PROCESSO SUDDIVISO IN *STADI* E BASATO SUL FATTO CHE:

- **OGNI ELEMENTO DELLO SPAZIO DI RICERCA È CONSIDERATO COSTITUITO DA DIVERSE *COMPONENTI* E AD OGNI STADIO VIENE SCELTA UNA COMPONENTE**
- **SE OGNI ELEMENTO DELLO SPAZIO DI RICERCA È STRUTTURATO IN n COMPONENTI, DOPO i STADI (i<n) SI È COSTRUITA UNA *SOLUZIONE PARZIALE*. LA CARATTERISTICA DELLA TECNICA DI BAKCTRACKING È CHE IN MOLTI PROBLEMI È POSSIBILE GIUNGERE ALLA CONCLUSIONE CHE LA SOLUZIONE PARZIALE GENERATA È FALLIMENTARE. RICONOSCIUTO CIO´ *L'ALGORITMO INTERROMPE IL PROCESSO DI COSTRUZIONE E TENTA ALTE VIE*.**

Backtracking

Spazio di ricerca ≡ albero di decisione

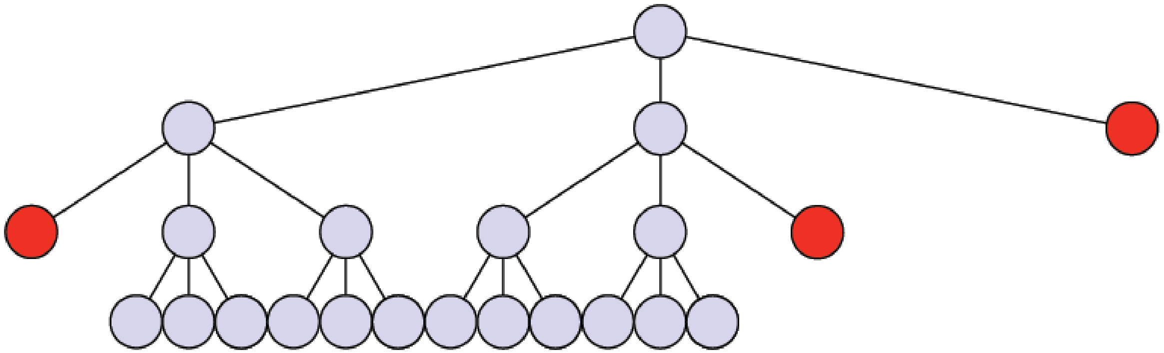
- Soluzioni ≡ foglie in un albero di decisione
- Soluzioni parziali ≡ nodi interni dell'albero di decisione
- Radice ≡ soluzione parziale vuota

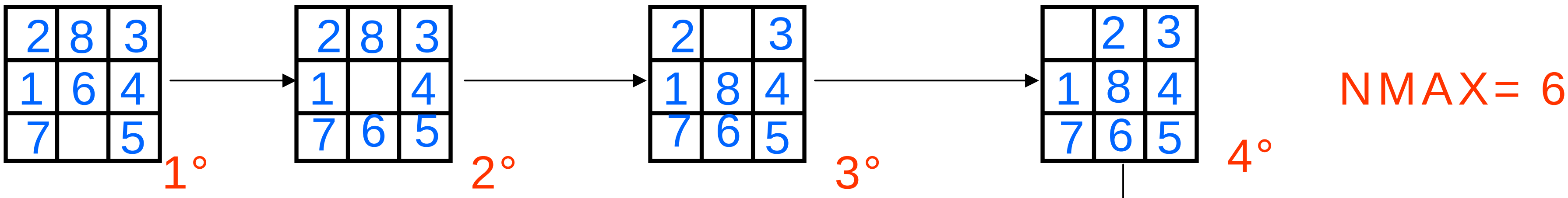


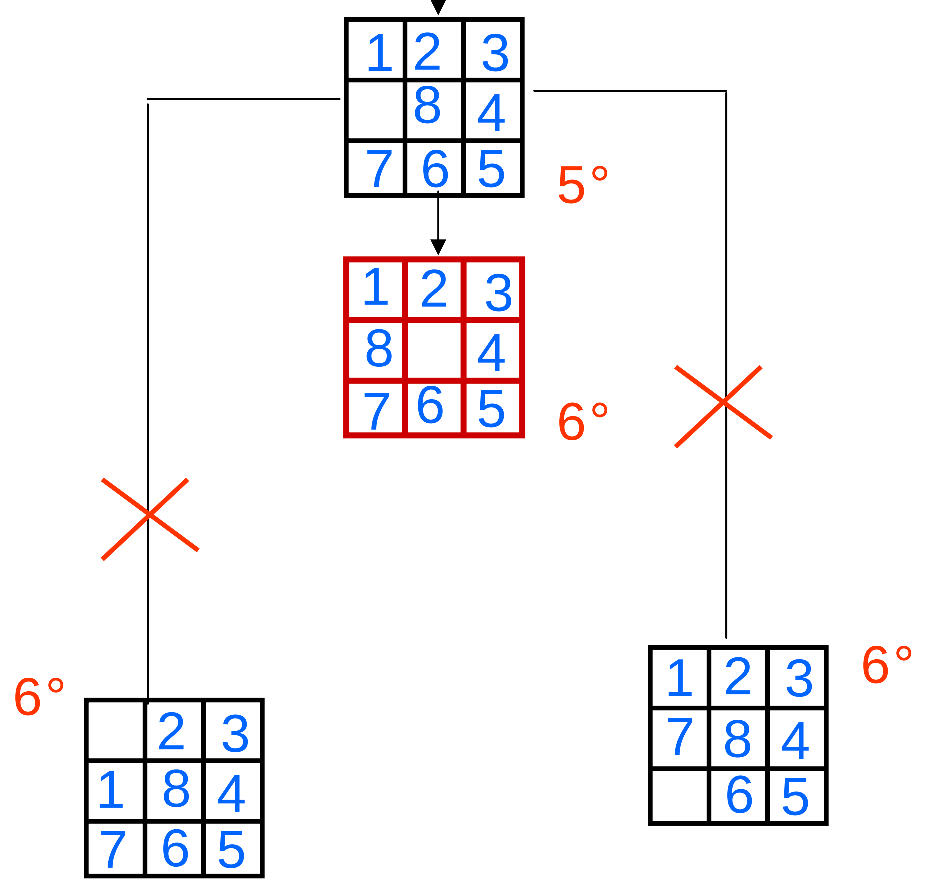
Backtracking

Lo spazio di ricerca può essere ridotto:

- "Rami" dell'albero che sicuramente non portano a soluzioni ammissibili possono essere "*potati*" (*pruned*)
- La valutazione viene fatta nelle soluzioni parziali radici del sottoalbero da potare







GIOCO DELL'OTTO CON TECNICA BACKTRACKING.

NELL'IPOTESI CHE LE MOSSE SIANO FATTE A CASO, IL RITORNO INDIETRO AVVIENE QUANDO:

- **SI GENERA UNA CONFIGURAZIONE GIA' PRODOTTA**
- **SI SONO GIA' FATTE PIU' DI UN NUMERO FISSATO DI MOSSE (NMAX) SENZA ARRIVARE ALLA SOLUZIONE**

ESEMPIO:

PROBLEMA DELLE N REGINE CON ALGORITMO DI BACKTRACKING.

COME SI È DETTO POSSIAMO ESPRIMERE LA SOLUZIONE ATTRAVERSO UNA PERMUTAZIONE DEI NUMERI DA 1 A N.

SIA N=8: LA SOLUZIONE SI PUÒ ESPRIMERE MEDIANTE IL VETTORE V DEFINITO ALIVELLO DI TIPO COME SEGUE:

```
TYPE    POSIZIONE=[1..8];
        VETTORE = ARRAY [1..8] OF POSIZIONE;
VAR  V:VETTORE;
```

I VINCOLI IMPOSTI SONO :

i. **LE COMPONENTI DI V DEVONO COSTITUIRE UNA PERMUTAZIONE DA 1 A 8; PER OGNI COPPIA DI INDICI i E j VALE $V[i] \neq V[j]$**

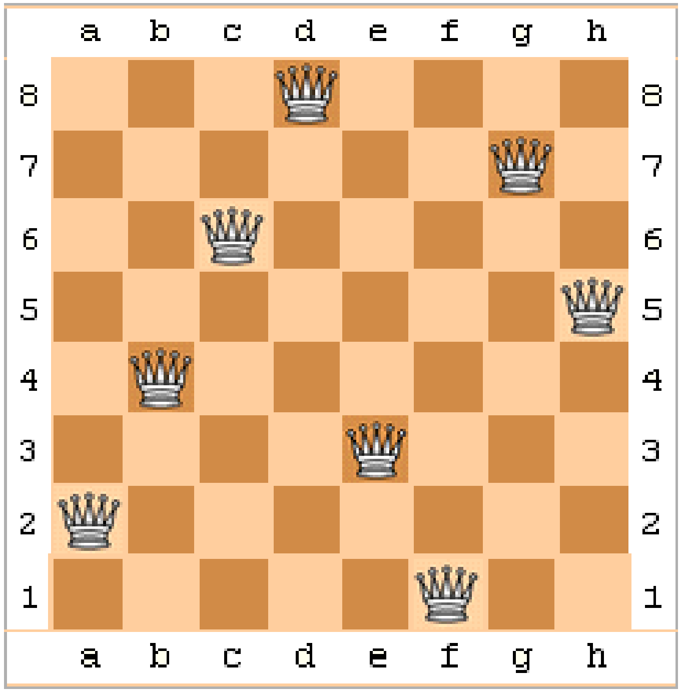
ii. **NON È POSSIBILE AVERE DUE ELEMENTI SULLA STESSA DIAGONALE, CIOE'**

$$(i - j) \neq (k - 1)$$

$$(i + j) \neq (k + 1)$$

IL SOTTOPROGRAMMA *VERIFICA* HA IL COMPITO DI ESAMINARE IL VETTORE V E SE IL VETTORE VIOLA UNO DEI VINCOLI RENDE *TRUE* LA VARIABILE a, ALTRIMENTI SE I VINCOLI SONO AMBEDUE SODDISFATTI a = *FALSE*.

UNA POSSIBILE SOLUZIONE



```
PROCEDURE  OTTO_REGINE ( V:VETTORE;
                                        SUCCESSO:BOOLEAN );
VAR   a,b:BOOLEAN ; k:INTEGER ;
BEGIN
    k ← 1; V[k] ← 1; SUCCESSO ← FALSE;
REPEAT
  VERIFICA(a); {PROGRAMMA CHE CONTROLLA IL VETTORE}
  IF (a) THEN    {SI CERCA UNA NUOVA POSIZIONE}
        REPEAT
          IF (V[k]<8) THEN BEGIN b ← TRUE;
                                 V[k] ← V[k] +1;
                             END
                      ELSE BEGIN {BACKTRACK}
                                 b ← FALSE;
                                 k ← k-1;
                             END;
        UNTIL   b   OR    (k=0)
```

```
   ELSE IF (k=8) THEN   SUCCESSO:=TRUE
          ELSE  {SI AGGIUNGE COMPONENTE}
             BEGIN
                    k ← k+1;
                    V[k] ← 1;
             END;
UNTIL   SUCCESSO   OR   (k=0)
END
```

ESEMPIO:

PROBLEMA DELLO STRING MATCHING (1)

TROVARE UNA OCCORRENZA DI UNA SEQUENZA P DI m CARATTERI ("PATTERN") IN UN'ALTRA SEQUENZA T DI n CARATTERI ("TESTO"). LE DUE STRINGHE P E T SONO FORMATE DA CARATTERI TRATTI DALLO STESSO ALFABETO ED m NON SUPERA n .

11101100011101111000 **T**

0110 **P**

UN ALGORITMO BANALE CONSISTE NEL CERCARE DI RICONOSCERE IL PATTERN (MODELLO) A PARTIRE DALLA PRIMA POSIZIONE DEL TESTO E SE IL PATTERN NON È INTERAMENTE RICONOSCIUTO, RIPETERE IL PROCEDIMENTO A PARTIRE DALLA POSIZIONE SUCCESSIVA NEL TESTO CONTINUANDO FINO AL RICONOSCIMENTO DELL'INTERO PATTERN O ALL'ESAURIMENTO DEL TESTO.

SE IL PATTERN NON È RICONOSCIUTO L'ALGORITMO "*TORNA*" A RIPETERE IL PROCEDIMENTO DALLA POSIZIONE CHE SEGUE.

LE STRINGHE P E T SI POSSONO REALIZZARE CON DUE VETTORI DI CARATTERI.

IL PROGRAMMA PUÒ REALIZZARSI MEDIANTE UNA FUNZIONE CHE RESTITUISCE LA POSIZIONE DI T A PARTIRE DALLA QUALE SI TROVA LA PRIMA OCCORRENZA DI P (SE IL PATTERN OCCORRE NEL TESTO) OPPURE n+1 SE P NON OCCORRE IN T.

```
FUNCTION      STRING1(T, P:VETTORE;
                          n ,m:INTEGER):INTEGER;
VAR i , j , k :INTEGER;
BEGIN
  i←1; j ← 1; k ← 1;
  WHILE (i ≤ n) AND (j ≤ m) DO
        IF T(i) = P(j) THEN
              BEGIN i ← i+1; j ← j+1; END
        ELSE
              BEGIN k ← k+1;i ← k; j ← 1 END;
   IF j >m THEN STRING1 ← k
   ELSE STRING1 ← i;
END
```

LA FUNZIONE EFFETTUA BACKTRACK SUGLI INDICI i E j

ESEMPIO:

STRING MATCHING (2)

(ALGORITMO DI KNUTH-MORRIS-PRATT)

TRAE VANTAGGIO DAI CONFRONTI GIA` FATTI PRECEDENTEMENTE SUL PATTERN.

INFATTI SE

T=110111011001

P=110110

I PRIMI 5 CARATTERI DI P SONO UGUALI AI PRIMI 5 DI T, SOLO IL SESTO È DIVERSO. AL MOMENTO DEL PRIMO BACKTRACK RISULTA i=j=6 E RIPARTIRE CON i=2 E j=1 È TRASLARE DI UNA POSIZIONE IL PATTERN RISPETTO AL TESTO.

VALE LA PENA DI VERIFICARE CHE NELLA SOTTOSEQUENZA RICONOSCIUTA DI P

11011 1

I PRIMI 4 CARATTERI NON COINCIDONO CON GLI ULTIMI QUATTRO, NÉ I PRIMI 3 CON GLI ULTIMI TRE

1101 1011

110 011

MA I PRIMI DUE COINCIDONO CON GLI ULTIMI DUE

11 11

NON VALE LA PENA DI RIPARTIRE CON i=2 O i=3 MA CON i=4 DIRETTAMENTE !!

POICHÉ I DUE CARATTERI A QUESTO PUNTO COINCIDONO CON QUELLI DI T

T= 110111011001

P= 11011 0

E’ CONVENIENTE RIPARTIRE COL BACKTRACK CON i=6 E j=3

6 È PROPRIO IL VECCHIO VALORE DI i AL MOMENTO DEL *BACKTRACK*, DUNQUE È INUTILE EFFETTUARE BACKTRACK SU i BASTA FARLO SU j PORTANDOLO DAL 6 ORIGINARIO ALL’ATTUALE 3.

IL PROCEDIMENTO

T = 1101110110 01 i=1 j=1

P = 110110 i =6 j=3

110110 IL SUCCESSIVO FALLISCE

110110 i=6 j=2

OK!

IL METODO

SI CONSIDERINO DUE “*COPIE*“ DEI PRIMI j-1 CARATTERI DEL PATTERN: SI DISPONGANO UNA SOTTO L’ALTRA IN MODO CHE IL PRIMO CARATTERE DELLA COPIA INFERIORE SIA ESATTAMENTE SOTTO IL SECONDO CARATTERE DELLA SUPERIORE.

SE TUTTI I CARATTERI SOVRAPPOSTI NELLE DUE COPIE NON SONO UGUALI, SI TRASLINO DI UNA POSIZIONE A DESTRA TUTTI I CARATTERI DELLA COPIA INFERIORE.

IL PROCEDIMENTO SI ARRESTA NON APPENA I CARATTERI SONO IDENTICI O QUANDO NON CI SONO PIÙ CARATTERI SOVRAPPOSTI.

IL NUOVO VALORE DI BACKTRACK DA ASSEGNARE A j SUCC(j) È UGUALE AL NUMERO DI CARATTERI SOVRAPPOSTI +1.

FORMALMENTE SI PONE PER j=1 SUCC(j)=0

```
PROCEDURE CALC_SUCC(P,SUCC:VETTORE; m:INTEGER);
VAR j,h:INTEGER;
BEGIN
    j ←; h ← 0; SUCC(1) ← 0;
    WHILE j ≤ m DO
        IF h=0 THEN BEGIN
                        j ← j+1; h ← 1;
                        IF j ≤ m THEN IF p(j)=P(1) THEN SUCC(j) ← 0
                                                   ELSE  SUCC(j) ← 1
                         END;
        ELSE IF P(j)=P(h) THEN
          BEGIN
                        j ← j+1; h ← h+1;
                        IF j≤m THEN
                          IF P(j)=p(h) THEN SUCC(j) ← SUCC(h)
                             ELSE SUCC(j) ← h;
           END
         ELSE
           h ← SUCC(h);
END;
```

```
FUNCTION STRING2(T, P:VETTORE; n ,m:INTEGER):INTEGER;
      VAR i , j :INTEGER; SUCC:VETTORE;
      BEGIN
        i ← 1; j ← 1;
        CALC_SUCC(P,SUCC,m);
        WHILE (i ≤ n) AND (j ≤ m) DO
                IF j=0 THEN
                   BEGIN
                    i ← i+1; j ← 1;
                   END
                ELSE
                   IF T(i)=P(j) THEN
                         BEGIN i ← i+1; j ← j+1; END
                  ELSE    j ← SUCC(j);
          IF j >m THEN STRING2 ← i – m
                         ELSE STRING2 ← i;
      END
```

TECNICA DI BACKTRACKING: ALBERO DI RICERCA

SIA P UN PROBLEMA E SIA DATO UN METODO PER ASSOCIARE AD OGNI ISTANZA i DI P UNO SPAZIO DI RICERCA Z_i.

SIA DEFINITO UN MODO PER STRUTTURARE OGNI ELEMENTO DI Z_i IN UN NUMERO FINITO DI COMPONENTI.

***L'ALBERO DI RICERCA* ASSOCIATO A i TRAMITE Z_i È UN ALBERO TALE CHE:**

- ***LA RADICE (LIVELLO 0) RAPPRESENTA UNA SOLUZIONE PARZIALE FITTIZIA***
- ***OGNI NODO INTERNO A LIVELLO j ∍' $j>0$ RAPPRESENTA UNA SOLUZIONE PARZIALE S IN CUI SONO STATE SCELTE LE PRIME j COMPONENTI, ED HA TANTI FIGLI QUANTI SONO i MODI POSSIBILI DI AGGIUNGERE LA $j+1$ ESIMA COMPONENTE AD S***
- ***OGNI FOGLIA È UN ELEMENTO DI Z_i***

PER UN **ALBERO DI RICERCA** ASSOCIATO AD UNA ISTANZA **i** TRAMITE Z_i È IMPORTANTE DEFINIRE:

- UN **METODO** PER RAPPRESENTARE OGNI **SOLUZIONE PARZIALE** (NODO DELL'ALBERO)
- UN **METODO** PER STABILIRE SE UNA SOLUZIONE PARZIALE VIOLA I **VINCOLI DEL PROBLEMA** (STABILITI DA R)
- UNA **FUNZIONE DI CONTROLLO DEL BACKTRACKING**

$$C: \text{NODI} \rightarrow \{ \text{TRUE} , \text{FALSE} \}$$

TALE CHE

- PER OGNI NODO **INTERNO N** DELL'ALBERO (soluzione parziale), **C(N)=TRUE** SE E SOLO SE LA SOLUZIONE PARZIALE ***VIOLA I VINCOLI***.
- SE **N** È UNA **FOGLIA** (soluzione totale), **C(N)=TRUE** QUANDO IL CORRISPONDENTE ELEMENTO DELLO SPAZIO DI RICERCA ***È NON AMMISSIBILE***.

SE **C(N)=TRUE** LA SOLUZIONE PARZIALE RAPPRESENTATA DAL NODO N DELL’ALBERO **NON PUO’** ESSERE COMPLETATA IN MODO DA COSTRUIRE UNA SOLUZIONE PER LA ISTANZA DEL PROBLEMA CONSIDERATA.

TUTTAVIA POICHE’ **C(N)** E’ CALCOLATA SULLE CARATTERISTICHE DEL SOLO NODO N, PUO’ ACCADERE CHE TALI CARATTERISTICHE NON SIANO ANCORA SUFFICIENTI A STABILIRE SE N PUO’ CONDURRE AD ALCUNA SOLUZIONE.

E’ POSSIBILE, CIOE’, CHE **C(N)=FALSE** MA CHE LA SOLUZIONE PARZIALE RAPPRESENTATA DA N **NON POSSA ESSERE COMPLETATA NEGLI STADI SUCCESSIVI** IN MODO DA OTTENERE LA SOLUZIONE TOTALE. QUESTO COSTRINGE AD UN RITORNO ALL’INDIETRO SULL’ALBERO DI BACKTRACKING **A PIU’ LIVELLI PRECEDENTI**

ALGORITMO DI BAKCTRACKING

L'ALGORITMO SI RIFERISCE ALL'ALBERO DI RICERCA B_i ASSOCIATO ALLA ISTANZA i DI UN PROBLEMA. SI COMPONE COME SE EFFETTUASSE UNA VISITA IN PROFONDITÀ:

1. OGNI VOLTA CHE NELLA VISITA SI ANALIZZA UN NODO SI APPLICA LA FUNZIONE DI CONTROLLO DEL BACKTRACKING AL NODO. SE LA FUNZIONE RESTITUISCE TRUE, ALLORA QUEL NODO E TUTTO IL SOTTOALBERO ASSOCIATO AL NODO VIENE ABBANDONATO E LA VISITA CONTINUA
2. SE IL PROBLEMA È DI RICERCA, LA VISITA TERMINA QUANDO SI INCONTRA UNA SOLUZIONE (UN NODO FOGLIA CON FUNZIONE DI BACKTRACKING = FALSE) OPPURE QUANDO NON ESISTONO ALTRI NODI DA VISITARE.

3. SE IL PROBLEMA È DI <u>OTTIMIZZAZIONE</u>, L'ALGORITMO UTILIZZA UNA VARIABILE ***OTTIMO CORRENTE***, CHE MEMORIZZA AD OGNI PASSO IL MIGLIOR ELEMENTO AMMISSIBILE. L'ALGORITMO NON SI INTERROMPE AL PRIMO ELEMENTO AMMISSIBILE TROVATO, MA CONTINUA LA VISITA AGGIORNANDO TALE VARIABILE

ESEMPIO:

PROBLEMA DEL PARTIZIONAMENTO DI UN INSIEME.

DATO UN INSIEME $Y = \{y_1, y_2, y_3, .. , y_n\}$ DI N INTERI POSITIVI LA CUI SOMMA È $2M$ VOGLIAMO SAPERE SE ESISTE UN SOTTOINSIEME DI Y LA CUI SOMMA SIA PARI A M.

DATA L'ISTANZA

$$Y = \{8, 5, 1, 4\}$$

CERCHIAMO UN SOTTOINSIEME X LA CUI SOMMA SIA 9.

LA SOLUZIONE PUÒ ESSERE ESPRESSA MEDIANTE UN VETTORE

$$[x_1, .. , x_4]$$

DOVE $x \in \{ 0 , 1 \}$ E $x_i=1$ IFF $y_i \in x$.

L'ALGORITMO DI *BACKTRACKRING* INVECE DI GENERARE TUTTE LE QUADRUPLE $[x_1, .. , x_4]$ COSTRUISCE IL VETTORE SOLUZIONE PARTENDO DAL VETTORE VUOTO E AGGIUNGENDO UNA COMPONENTE ALLA VOLTA.

IL COMPORTAMENTO DELL' ALGORITMO SI PUÒ DESCRIVERE MEDIANTE LA RAPPRESENTAZIONE AD ALBERO DELLO SPAZIO DI RICERCA.

LA RADICE RAPPRESENTA IL VETTORE VUOTO.

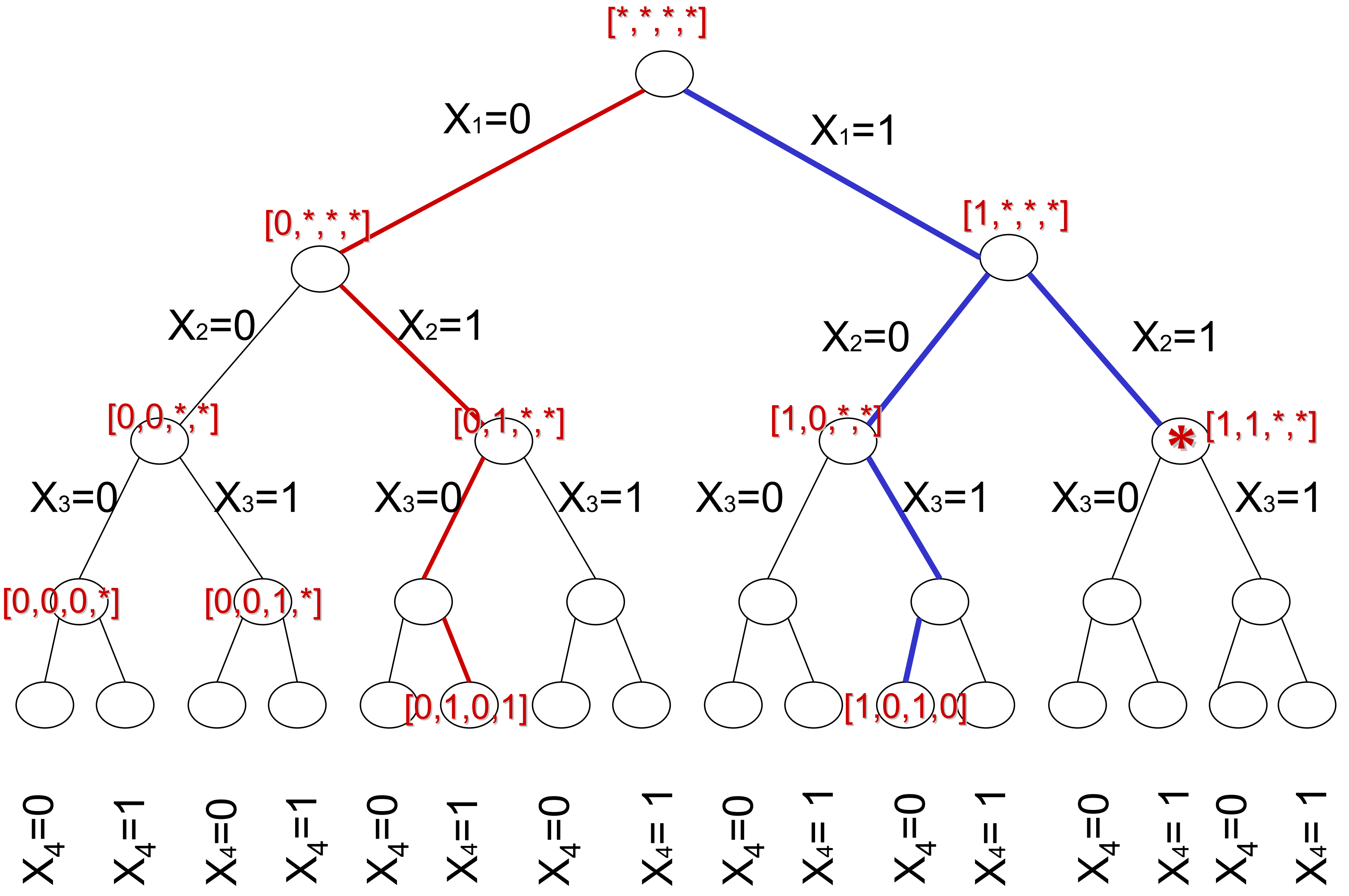
A LIVELLO 1 ABBIAMO I VETTORI CHE È POSSIBILE FORMARE CON LA PRIMA VARIABILE $x_1 = 1$ E $x_1 = 0$, E COSI' VIA.

SOLO NELLE FOGLIE AVRO' LA CONFIGURAZIONE COMPLETA DEL VETTORE E SOLO ALCUNE DELLE FOGLIE RAPPRESENTANO SOLUZIONI AMMISSIBILI.

L'ALGORITMO DI *BACKTRACKING* E' UN ALGORITMO DI VISITA IN ORDINE ANTICIPATO DELL'ALBERO DI RICERCA.

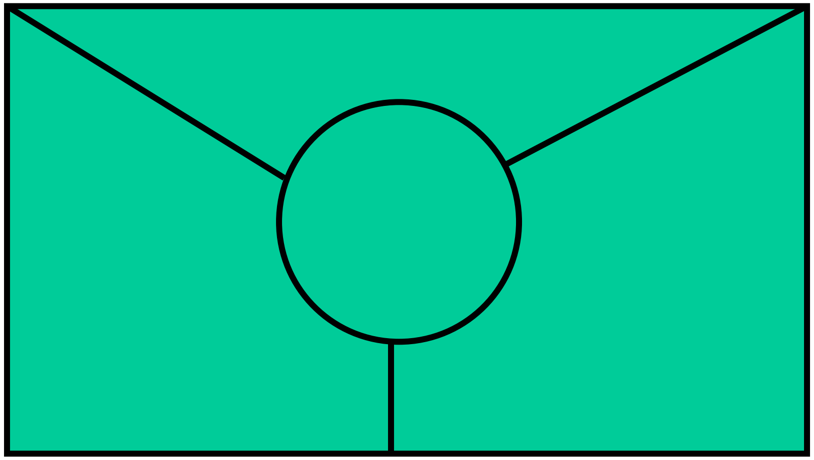
IL NODO SEGNATO CON * NELLA FIGURA CHE SEGUE RAPPRESENTA L'INSIEME DELLE POSSIBILI SOLUZIONI IN CUI $x_1 = 1$ E $x_2 = 1$; POICHÉ 8+5=13 SU QUESTO CAMMINO CERTO NON SI TROVERÀ SOLUZIONE.

ALBERO DI RICERCA PER IL PROBLEMA DEL PARTIZIONAMENTO DI Y={8,5,1,4}. SUI NODI SONO LE CONFIGURAZIONI DI X GENERATE

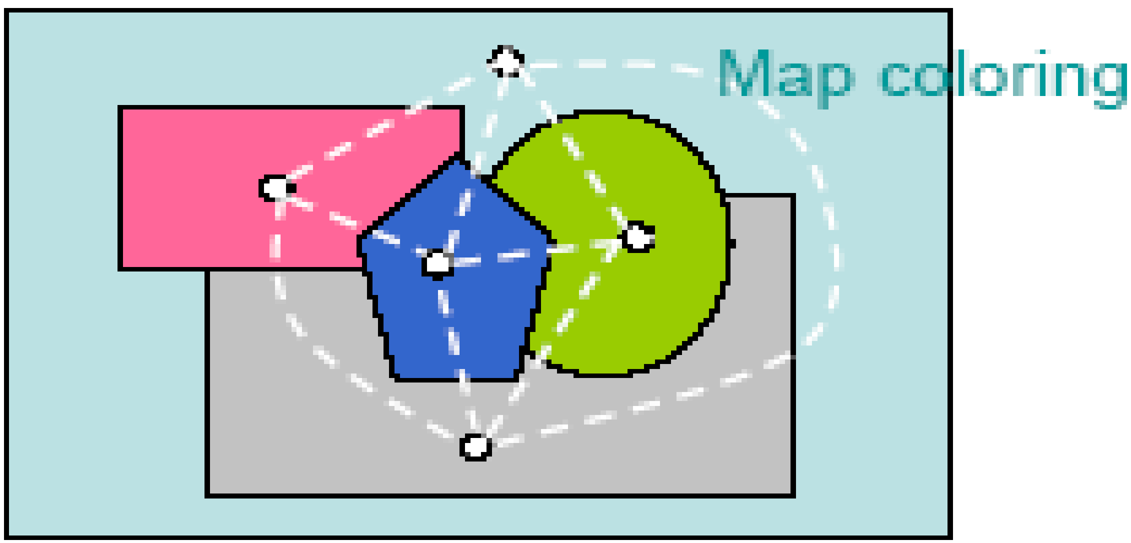


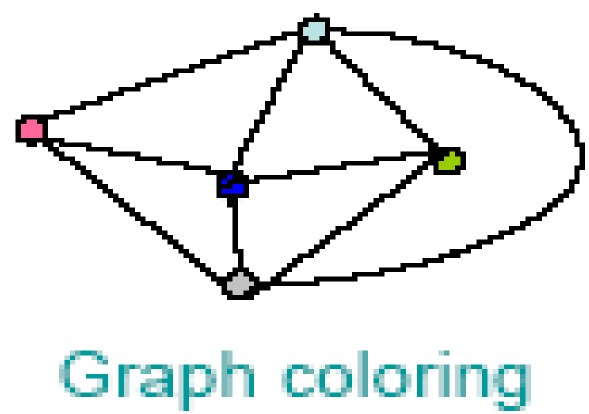
Il problema della colorazione di mappe

Data una mappa con **n** regioni ed una palette di **c** colori quanti modi ci sono per colorare la mappa così che nessuna regione abbia il medesimo colore delle regioni confinanti?



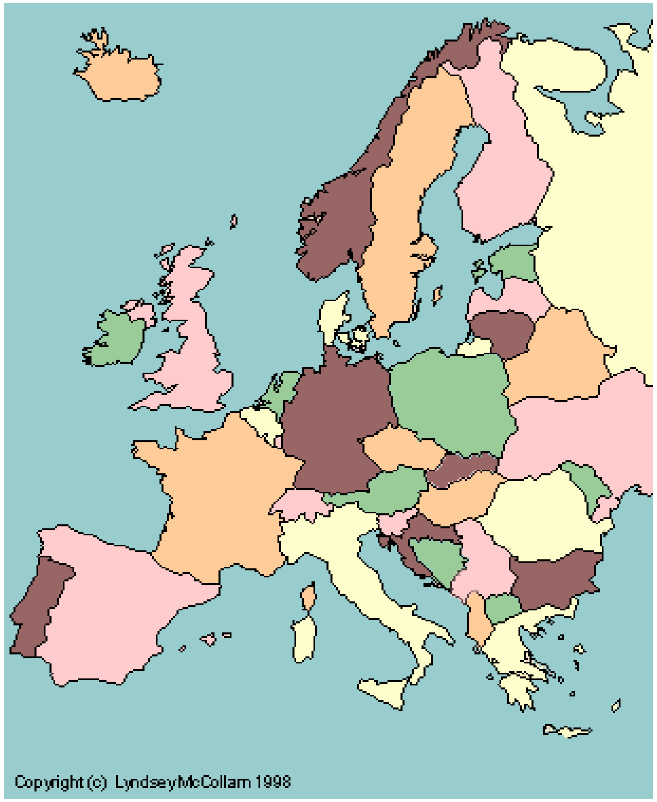
Il problema della colorazione di mappe





Il problema della colorazione di mappe

- Colora le nazioni con il minor numero di colori
- I colori devono essere differenti per regioni adiacenti
- 4 colori bastano sempre



Il metodo backtracking

- Per colorare una mappa con non più di 4 colori:
  - colora(regione n):
    - Se tutte le regioni sono state colorate (n > numero di regioni) return *successo*; altrimenti,
    - Per ogni colore c dei quattro colori,
      - Se la regione n non è adiacente a una regione che è stata colorata con c
        - » Colora la regione n col colore c
        - » Ricorsivamente colora la regione n+1
        - » Se va bene, return *successo*
    - Se si forma un ciclo, return *fallimento*

ESEMPIO

IMPOSTAZIONE DELL'ALGORITMO DI BACKTRACKING PER RISOLVERE IL PROBLEMA DELLO ZAINO

- L'ALBERO DI RICERCA SI OTTIENE CONSIDERANDO CHE, SE n SONO GLI OGGETTI, OGNI ELEMENTO DELLO SPAZIO DI RICERCA SI PUO' COSTRUIRE IN n STADI DOVE ALL' *i-ESIMO STADIO* SI DECIDE SE INCLUDERE O NO, NEL SOTTOINSIEME RAPPRESENTATO DA X, l'*OGGETTO i-ESIMO*. L'ALBERO DI RICERCA E' UN ALBERO BINARIO DI PROFONDITA' n;
- OGNI SOLUZIONE PARZIALE E' RAPPRESENTATA CON UN VETTORE E UN INDICE (*STADIO DEL PROCESSO*): L'ELEMENTO i-ESIMO DEL VETTORE VALE 1 SE E SOLO SE IL i-ESIMO OGGETTO E' NELLO ZAINO;

ESEMPIO /2
IMPOSTAZIONE DELL'ALGORITMO DI BACKTRACKING PER RISOLVERE IL PROBLEMA DELLO ZAINO

- LA **FUNZIONE DI CONTROLLO DEL BACKTRACKING** SU UN NODO DELL'ALBERO, CUI CORRISPONDE LA SOLUZIONE PARZIALE <$X_1,\ldots,X_l$>, DEVE RESTITUIRE IL VALORE **TRUE** SE LE SCELTE FATTE PORTANO AD UN COSTO TOTALE GIA' MAGGIORE DEL BUDGET CIOE'

$$\sum_{i=1..l} C_i X_i > B$$

LA FUNZIONE, INOLTRE, VALE TRUE SE IL MASSIMO PROFITTO OTTENIBILE CON LE SCELTE GIA' FATTE E' MINORE DEL PROFITTO P_{ott} CORRISPONDENTE ALL'OTTIMO CORRENTE

$$\sum_{(I=1..l)} P_i X_i + \sum_{(i=l+1,..N)} P_i < P_{ott}$$