Uvod u računarske algoritme

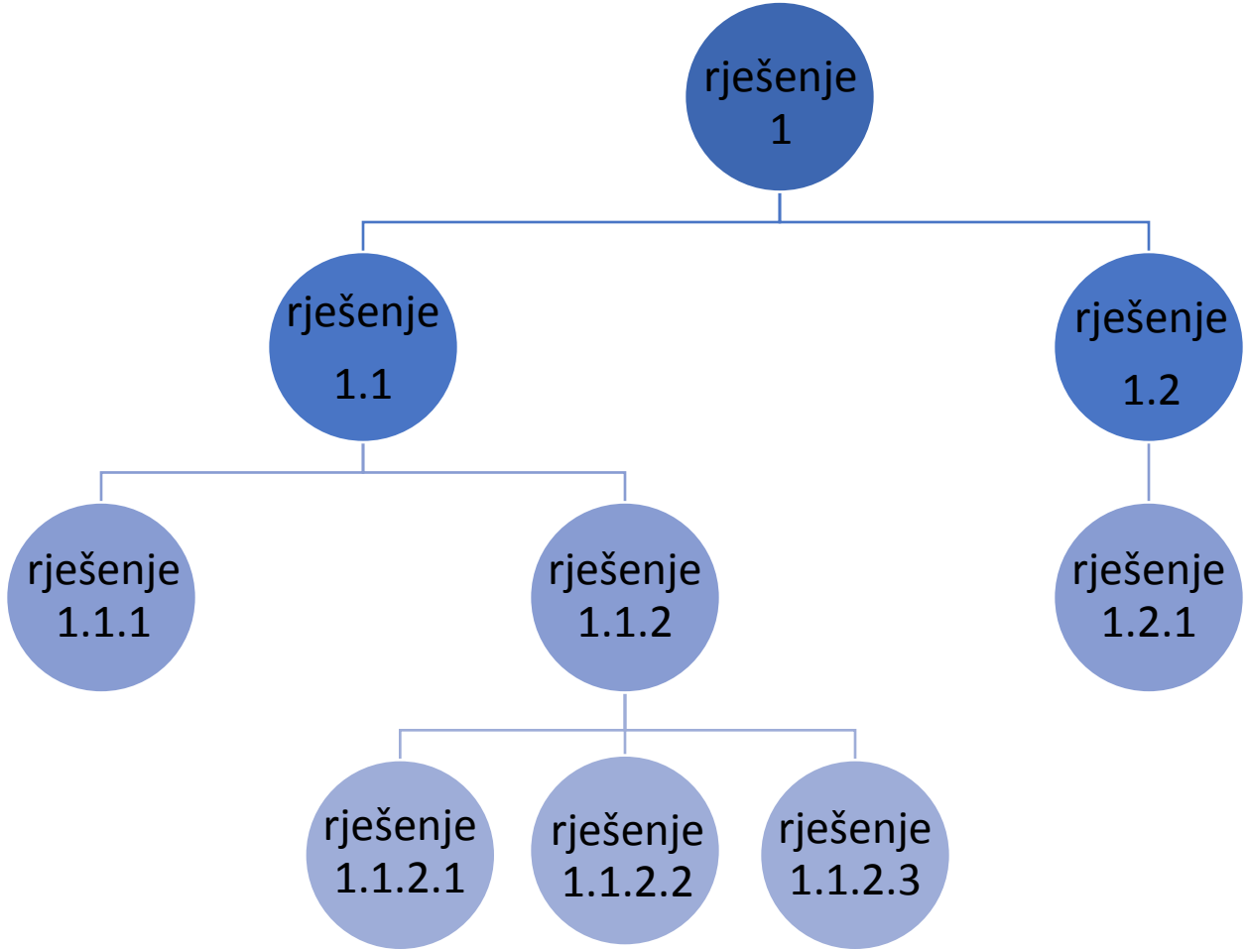
Backtracking rekurzija

Backtracking

- metod obrnute pretrage
- *backtracking* algoritmi traže rješenje (jedno ili više njih) problema u velikom skupu potencijalnih rješenja
- poboljšanje *brute force* pristupa (pristup grube sile)
  - Metod grube sile se zasniva na ispitivanju svih mogućih kombinacija ulaznih podataka u cilju pronalaska odgovarajućeg rješenja.
- inkrementalno gradi listu mogućih rješenja
- eliminiše velik broj kandidata za rješenje u svakom koraku
- rekurzivna implementacija
- potencijalna rješenja problema organizovana kao stablo (tzv. stablo pretrage potencijalnih kandidata)

Backtracking

- Kandidati za rješenje problema su čvorovi stabla pretrage



Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:

Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:

Trenutni rezultat	Min suma
-	2

Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:

Trenutni rezultat	Min suma
-	2

Trenutni rezultat	Min suma
1	2

Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:

Trenutni rezultat	Min suma
-	2

Trenutni rezultat	Min suma
1	2

Trenutni rezultat	Min suma
2	3

Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:

Trenutni rezultat	Min suma
-	2

Trenutni rezultat	Min suma
1	2

Trenutni rezultat	Min suma
2	3

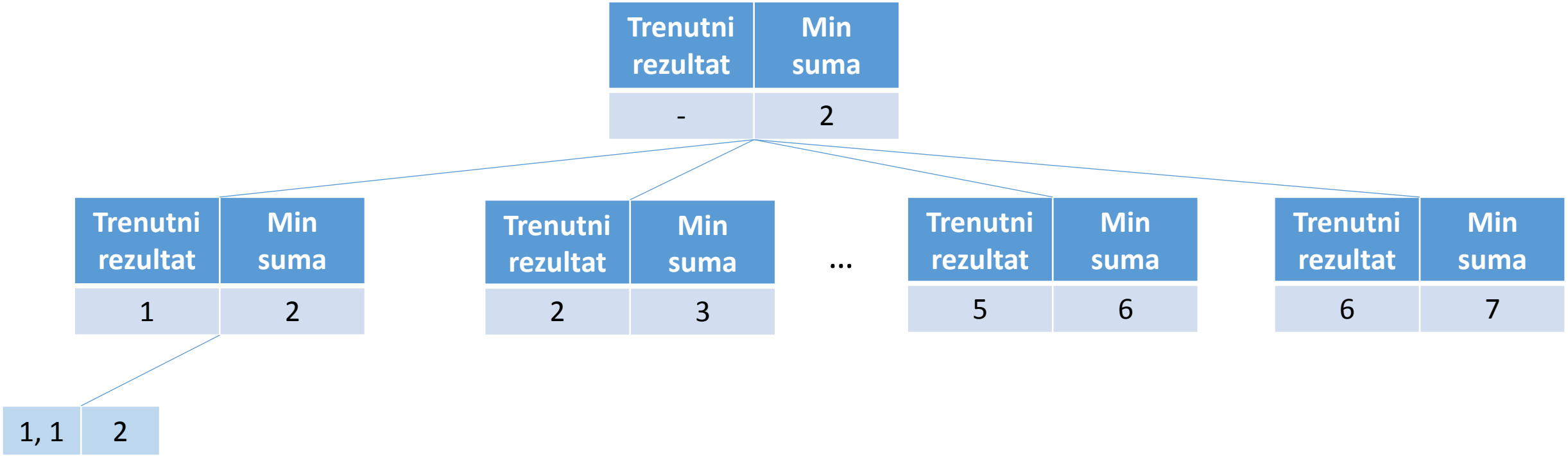
...

Trenutni rezultat	Min suma
5	6

Trenutni rezultat	Min suma
6	7

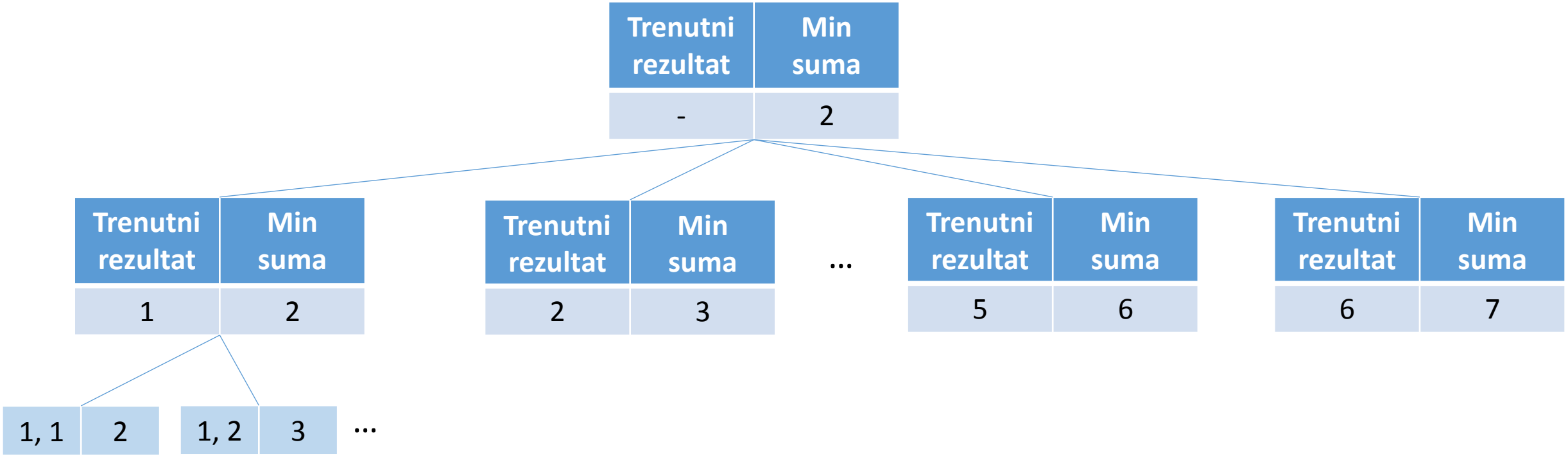
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



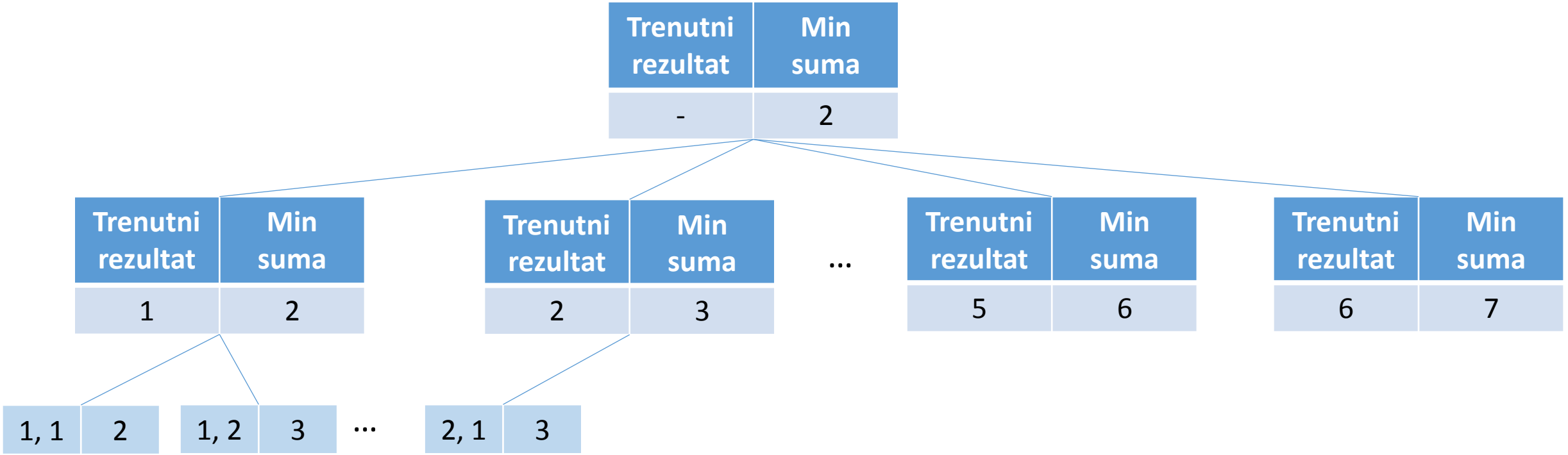
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



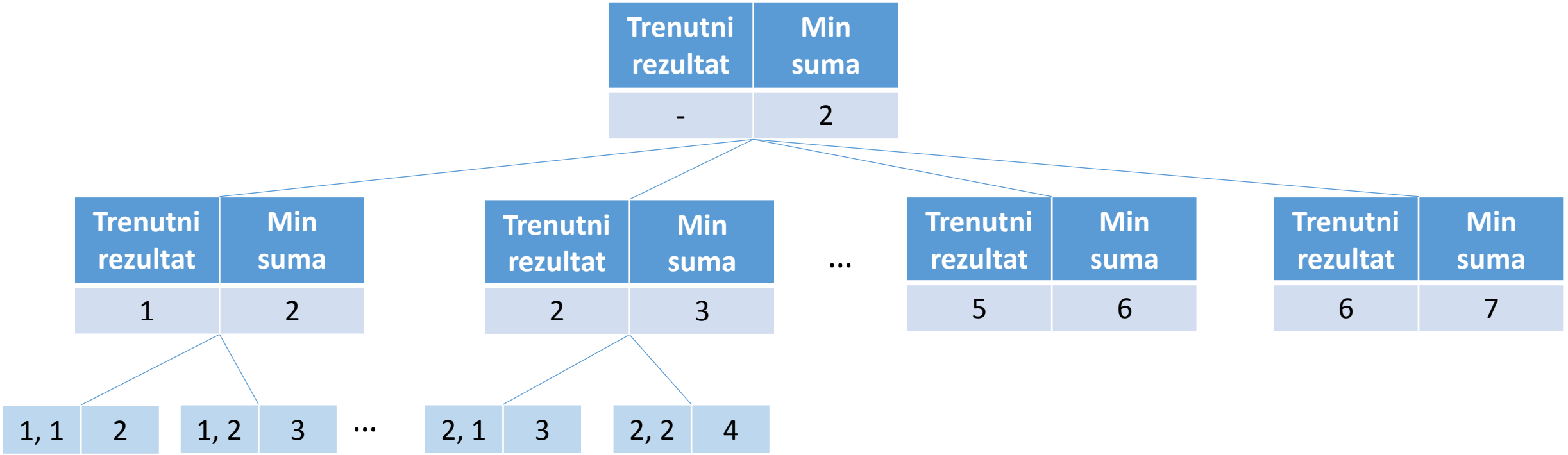
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



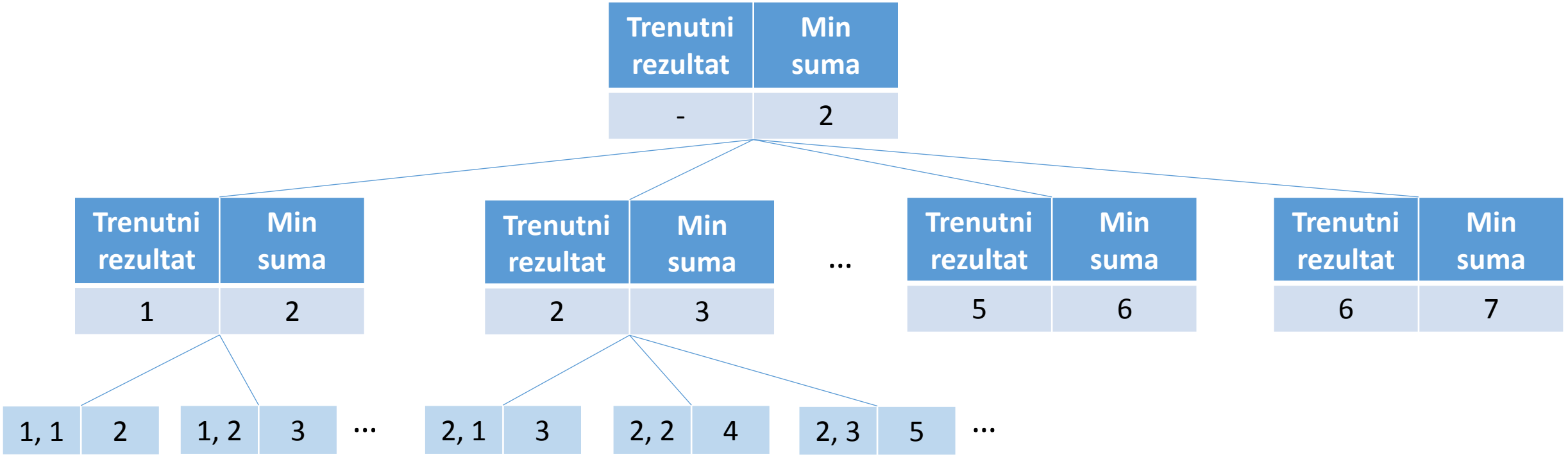
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



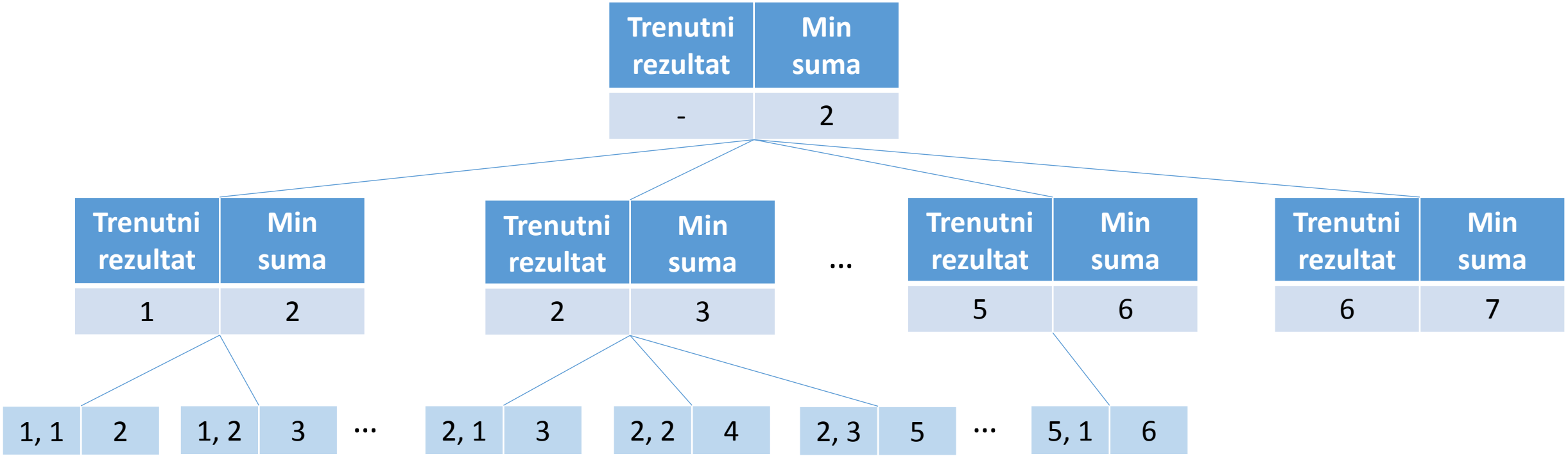
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



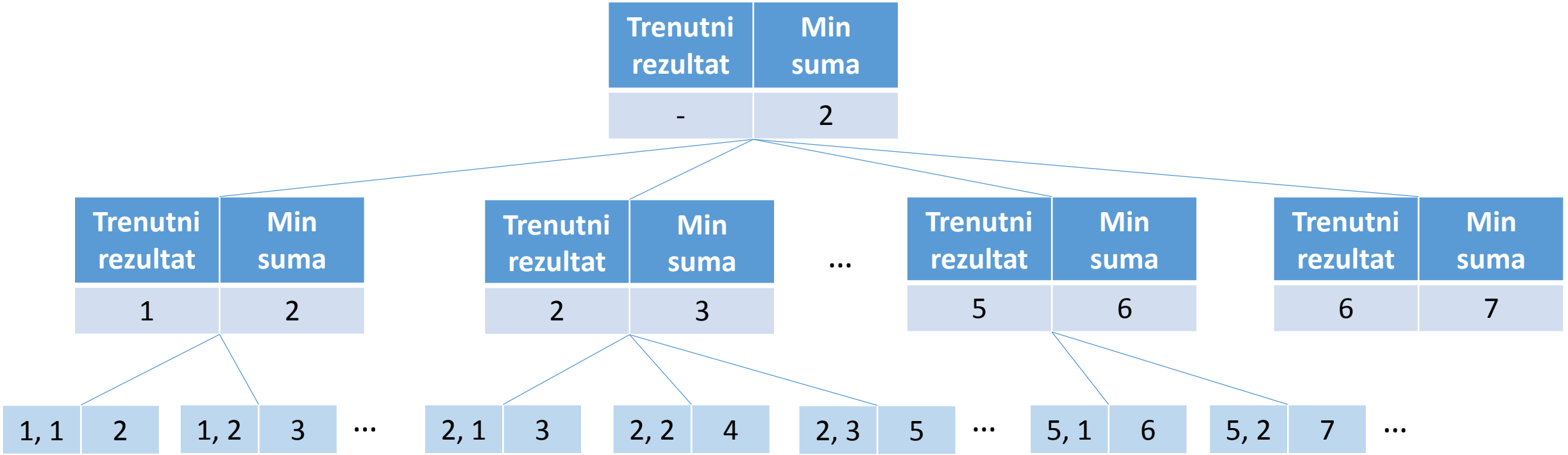
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



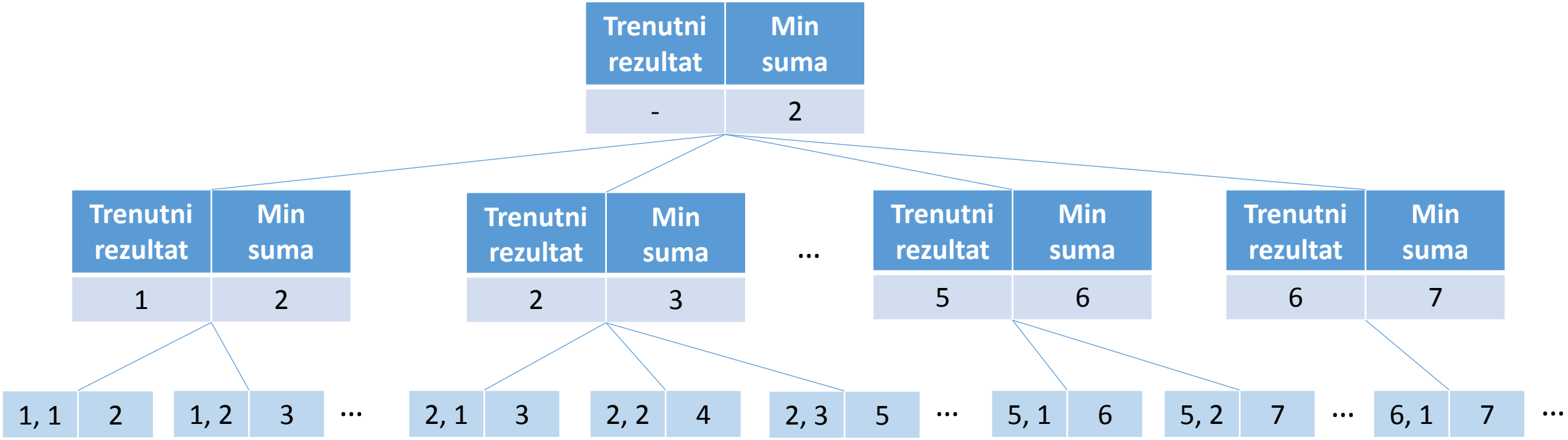
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



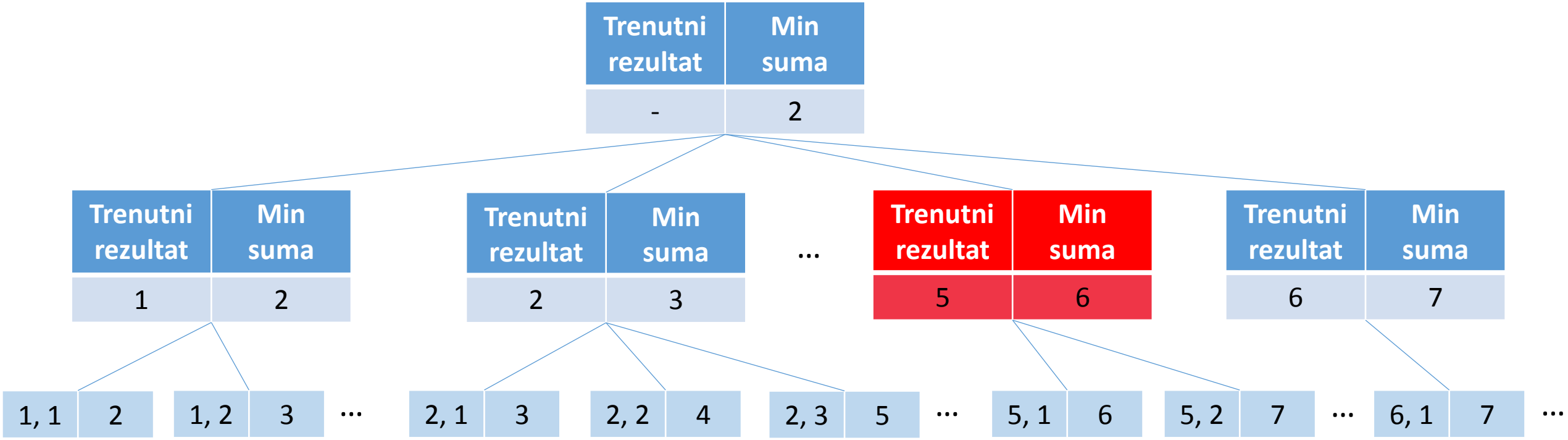
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



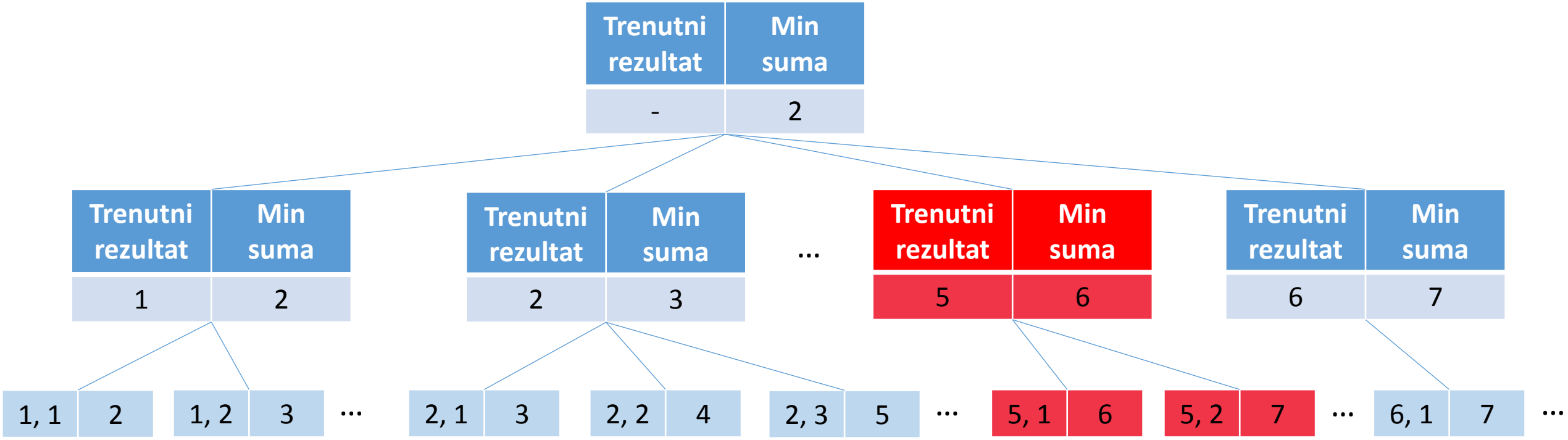
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



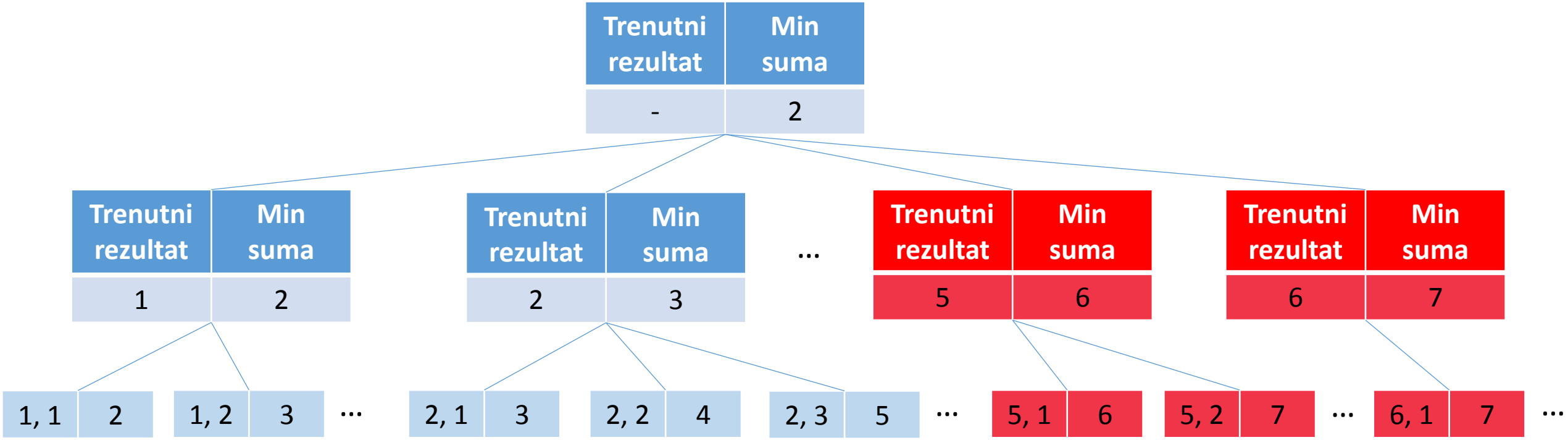
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



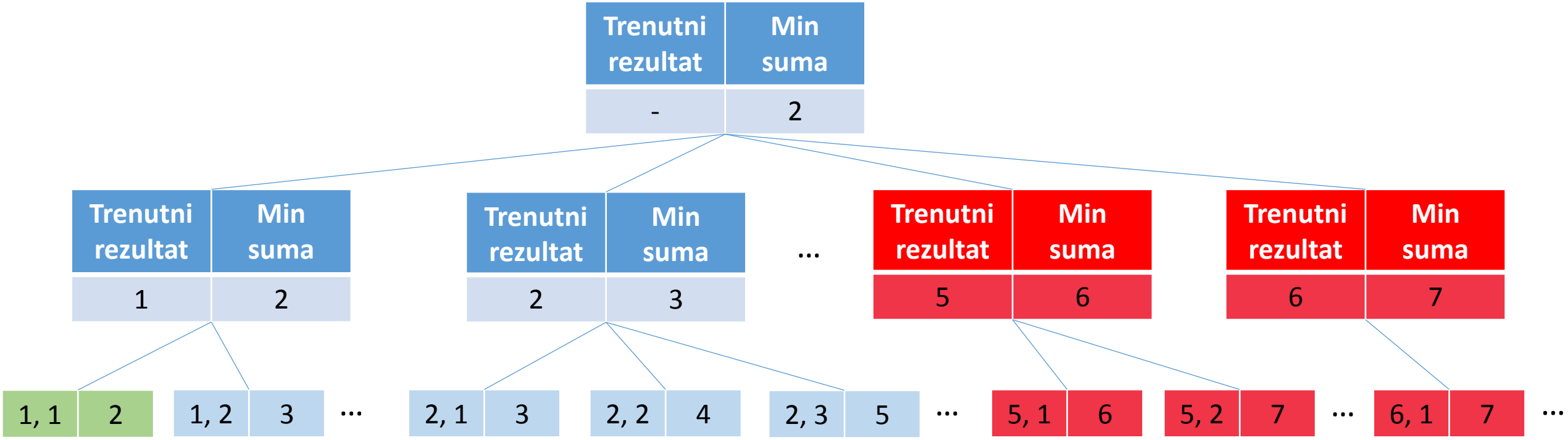
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



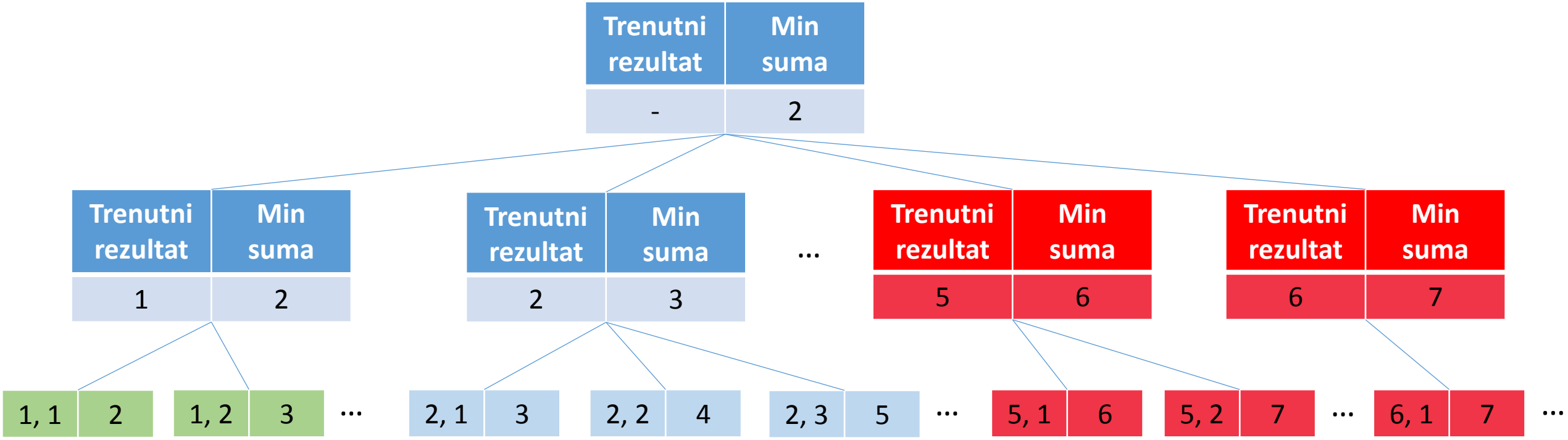
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



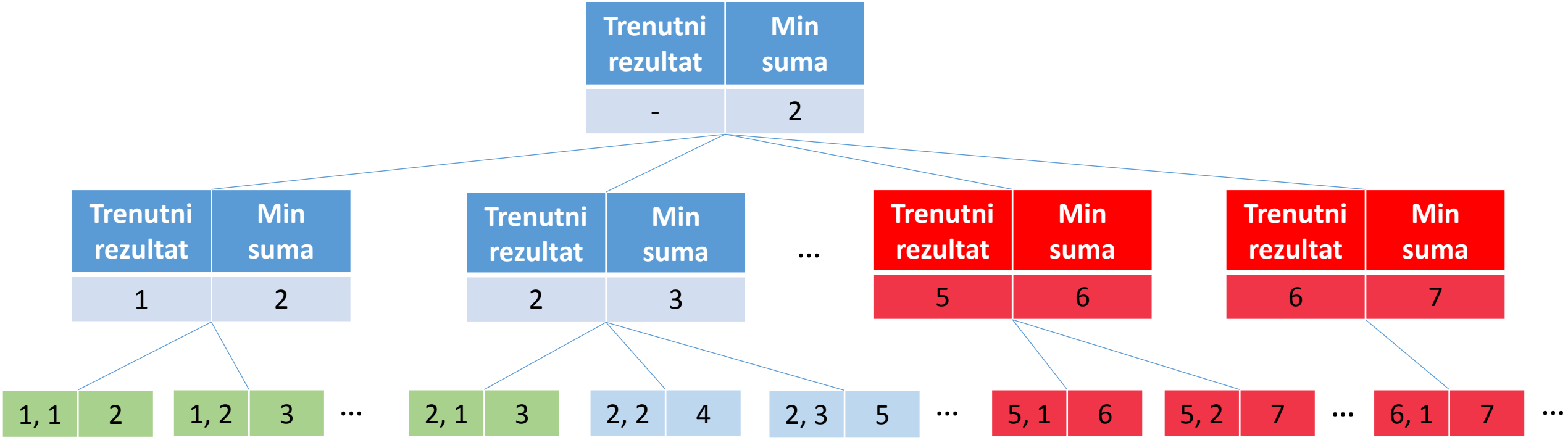
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



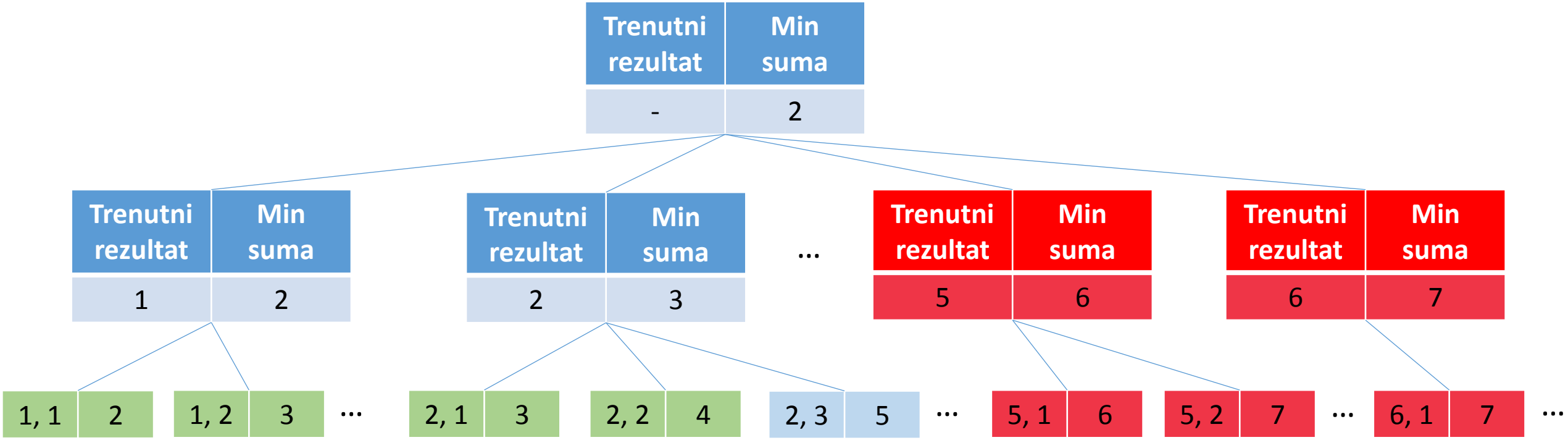
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



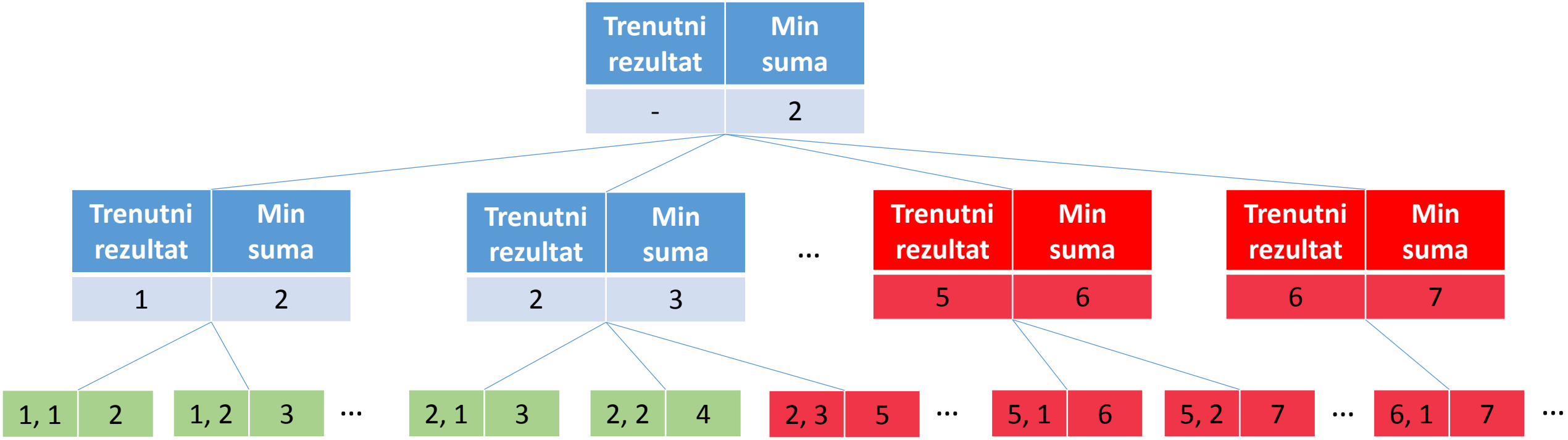
Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



Backtracking – primjer 1

- Formirati stablo pretrage za sljedeći problem:
  - Kocka se baca n puta. Realizovati algoritam koji ispisuje sve kombinacije rezultata tih bacanja za koje je ukupna suma manja od s.
- Za n=2 i s=5:



Backtracking – opšta implementacija

- *backtracking* algoritmi se obično implementiraju rekurzivno
- opšti oblik rekurzivnog *backtracking*a za pronalazak **svih rješenja** problema (pseudo kod):

```
void pronadjiRjesenja(n, drugi_parametri) {
   if (rjesenje_pronadjeno) {
      prikaziRjesenje();
   } else {
      for (val = od_prvog_do_zadnjeg_slucaja_za_n) {
         if (validnoRjesenje(val, n)) {
            uradiNesto(val);
            pronadjiRjesenja(n + 1, drugi_parametri);
         }
      }
   }
}
```

Backtracking – primjer 1

- Implementacija
  - int a[] – niz u koji snimamo rezultate pojedinačnih bacanja
  - int rbBacanja – redni broj bacanja koje trenutno obrađujemo
  - int ukupnoBacanja – ukupan broj izvršenih bacanja kocke
  - int trenutnaSuma – suma rezultata svih bacanja do tog trenutka

```
void bacanjeKocke (int a[], int rbBacanja, int ukupnoBacanja, int trenutnaSuma){
    if (rbBacanja == ukupnoBacanja){
        ispisi(a, ukupnoBacanja);
    }
    else{
        for(int j=1; j<=6; j++){
            if(trenutnaSuma<5 && trenutnaSuma+j<5){
                a[rbBacanja]=j;
                bacanjeKocke(a, rbBacanja+1, ukupnoBacanja, trenutnaSuma+j);
            } else return;
        }
    }
}
```

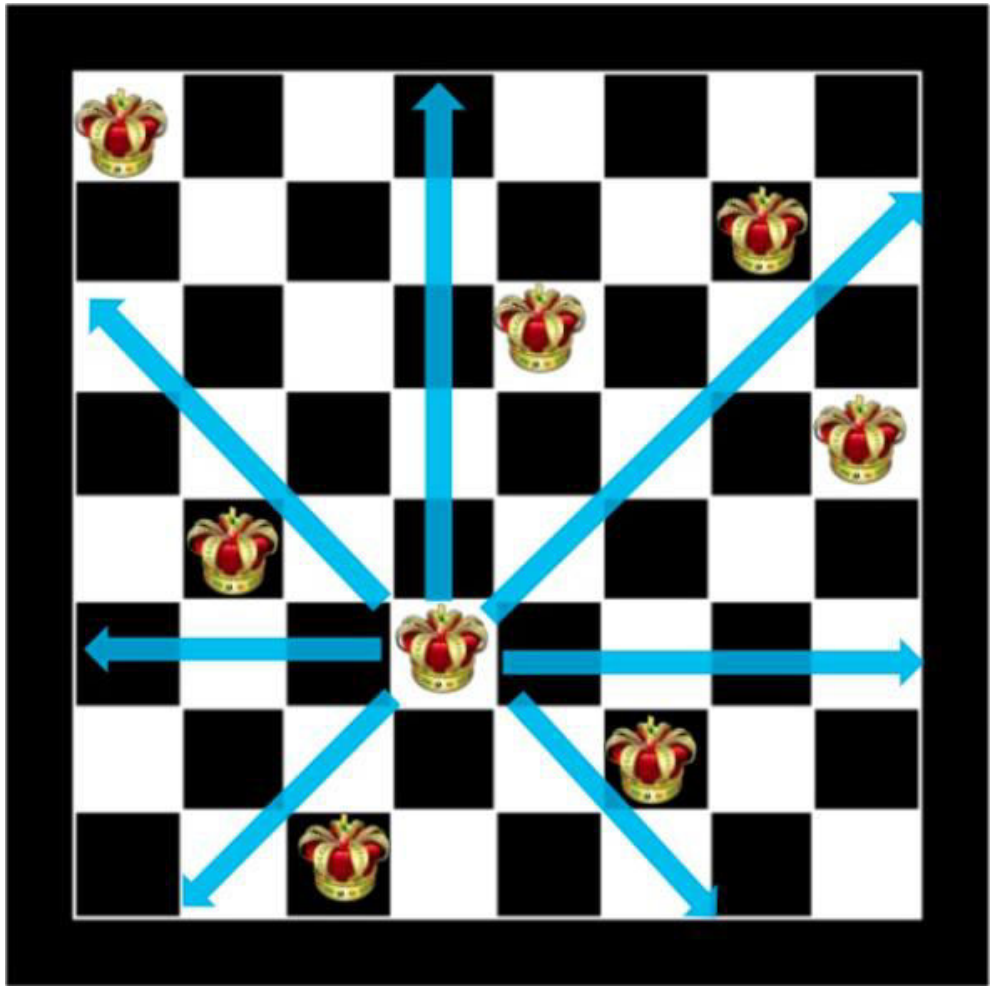
Backtracking

- tipični problemi koji se rješavaju *backtracking* rekurzijom:
  - slagalice (ukrštenice, Sudoku…)
  - problem 8 kraljica
  - ispitivanje zadovoljenja proizvoljne funkcije
  - određivanje kombinacija i permutacija
  - algoritmi za parsiranje teksta

- Problem 8 kraljica
  - traga se za rasporedom osam kraljica na standardnoj šahovskoj tabli, tako da nijedna ne napada bilo koju drugu
  - problem se može generalizovati na problem n kraljica

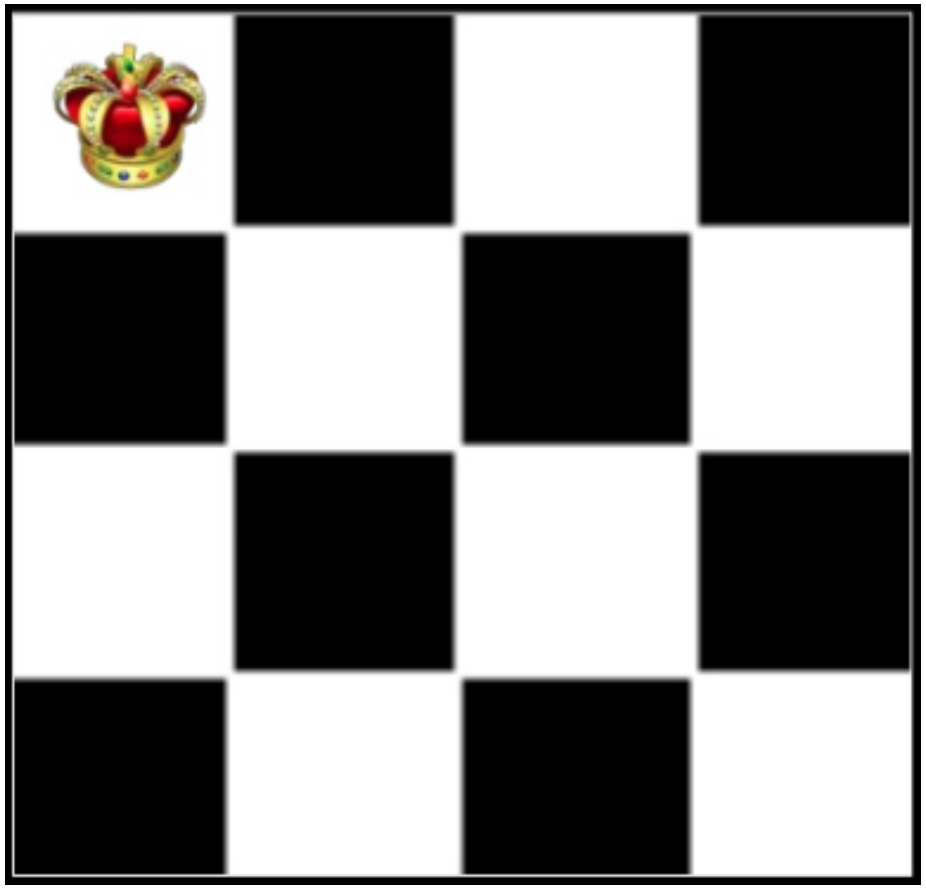
Backtracking – problem n kraljica

- Generalni pristup je sljedeći:
  - postaviti n kraljica u n različitih kolona ploče
  - *backtracking* algoritmom odrediti sve kombinacije pozicija kraljica, tj. indekse redova za svaku kolonu
  - odbaciti sva rješenja koja sadrže dvije pozicije koje omogućavaju međusobno napadanje kraljica, jer ne mogu voditi tačnom rješenju



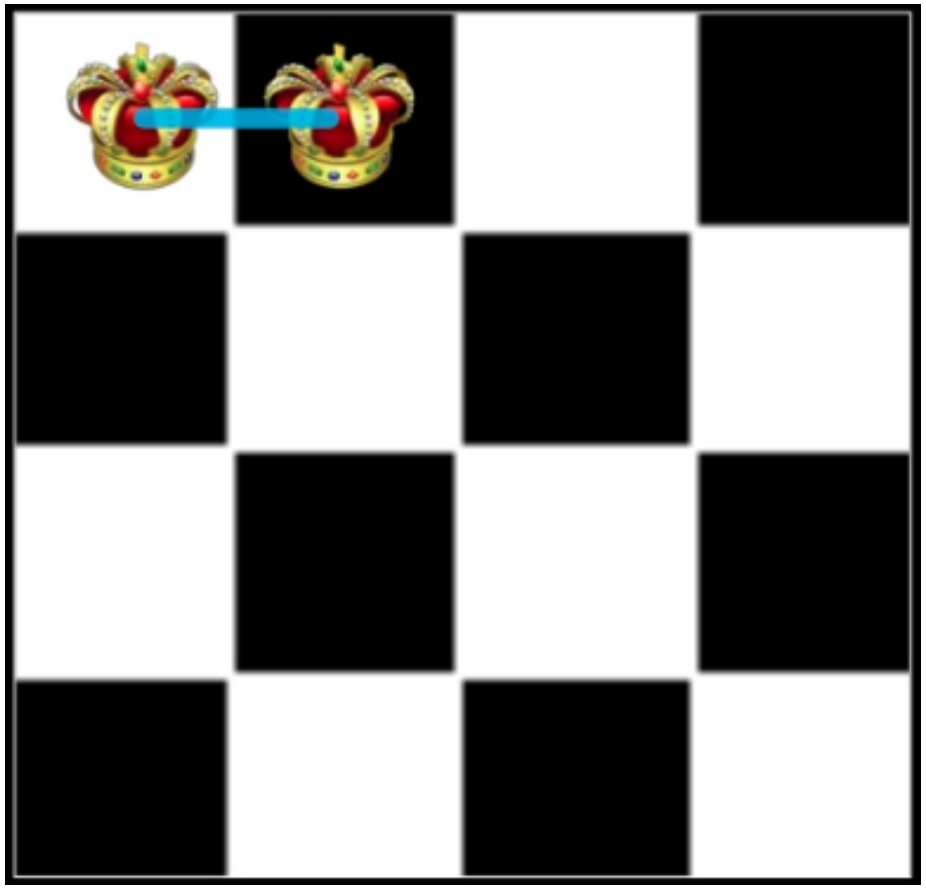
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



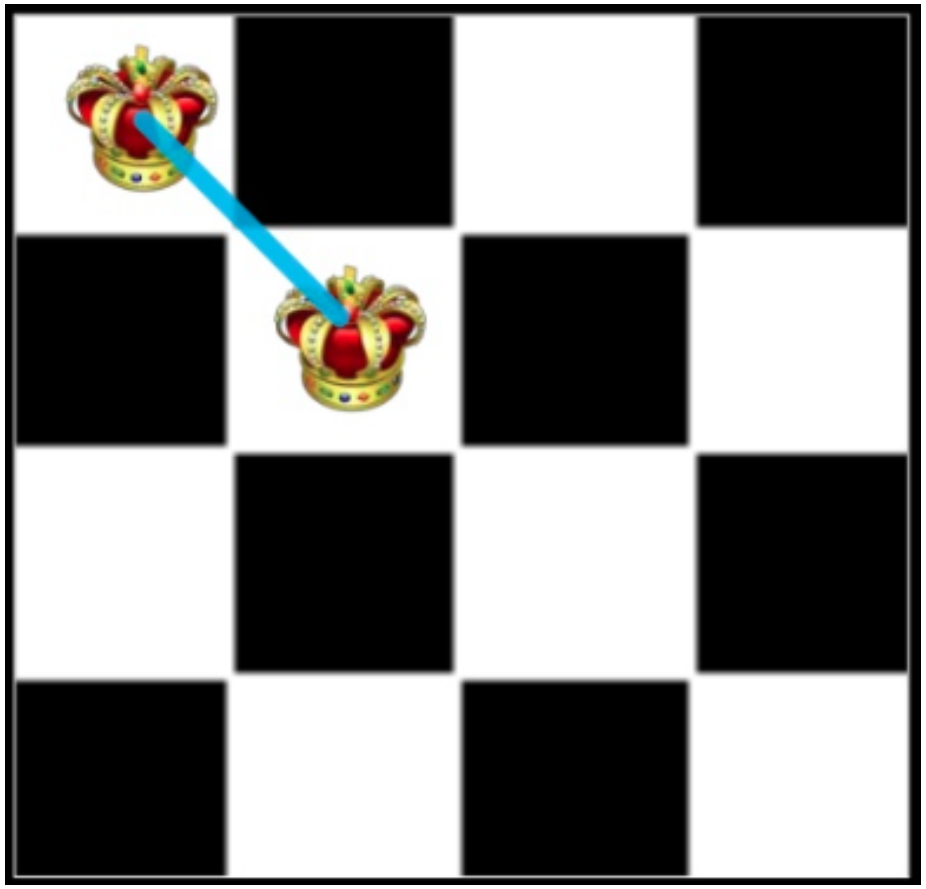
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



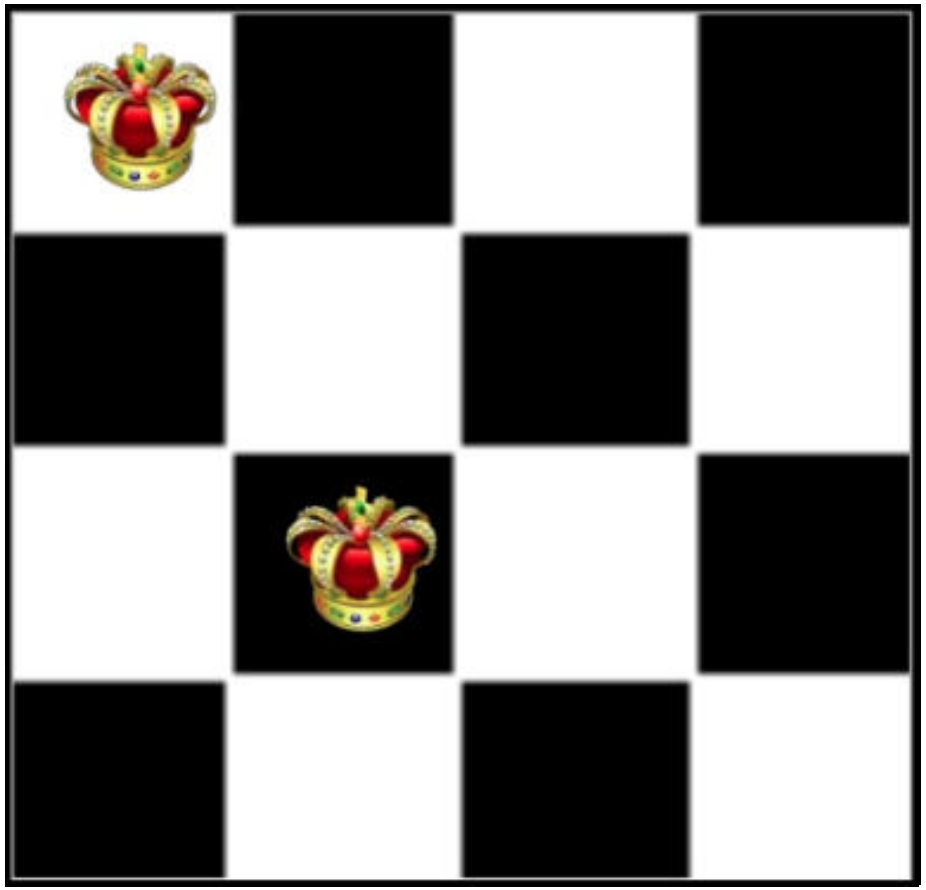
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



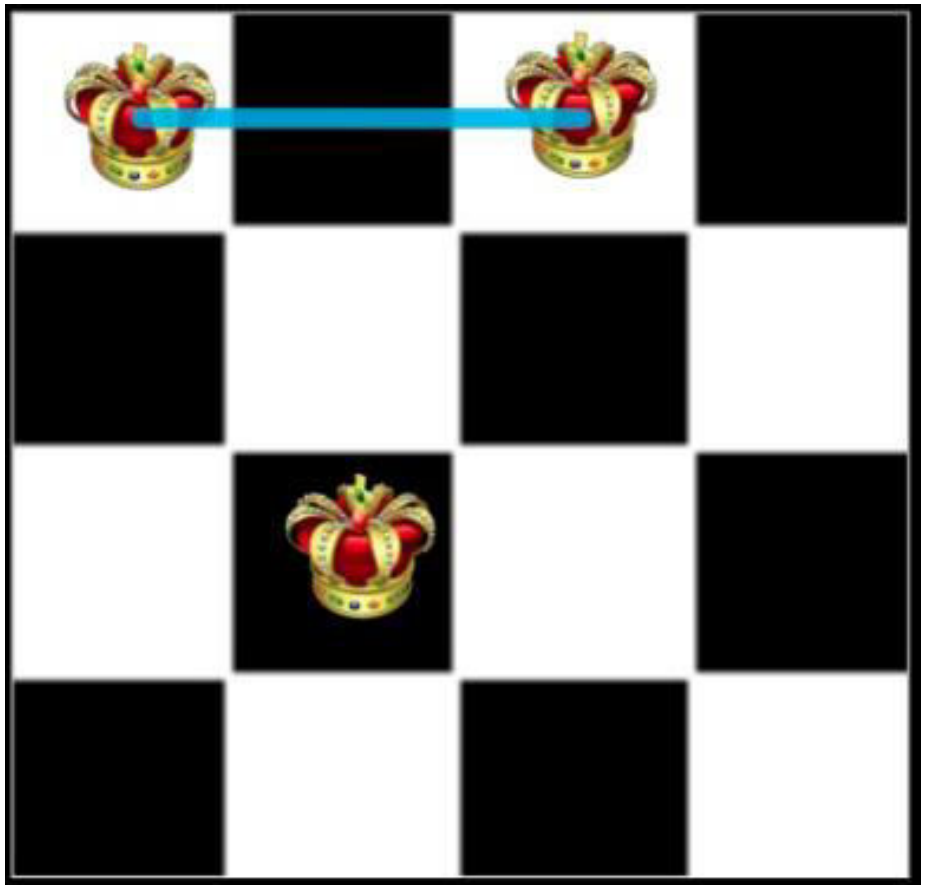
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



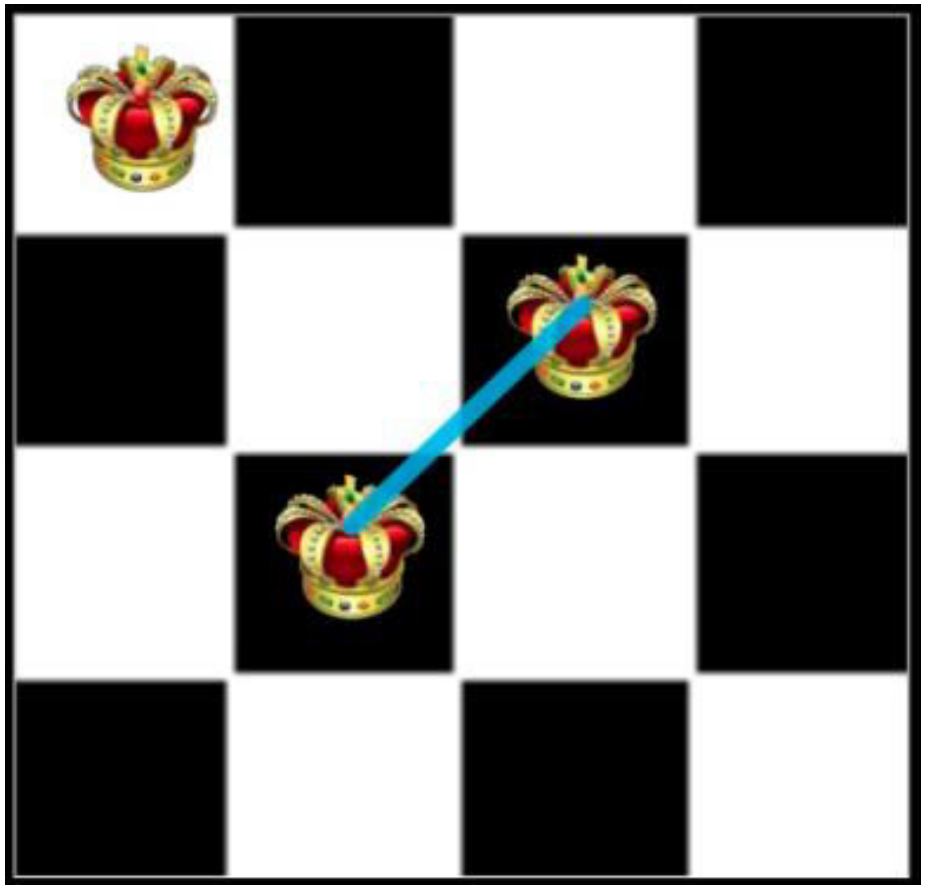
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



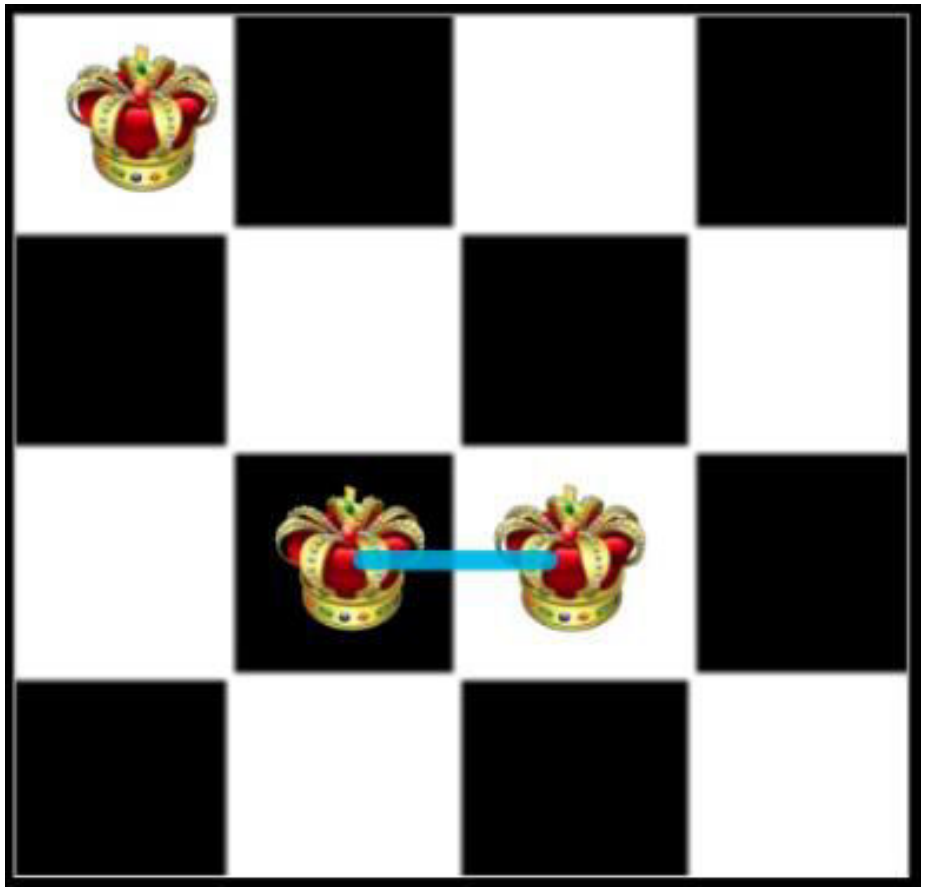
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



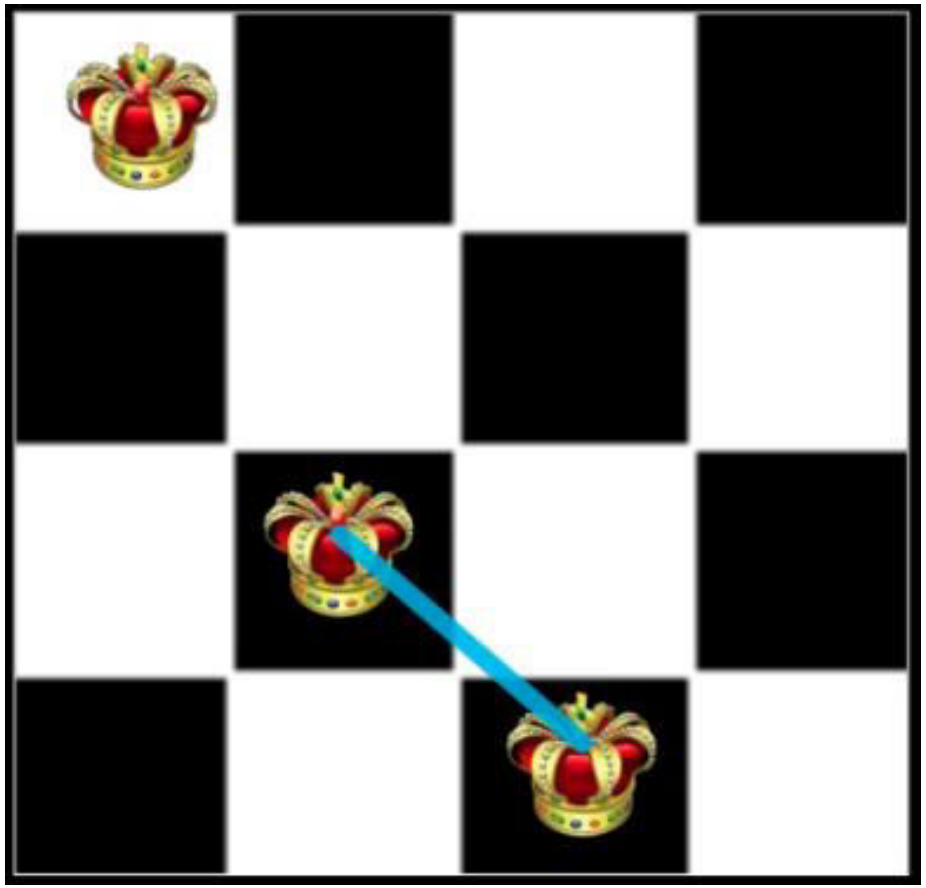
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



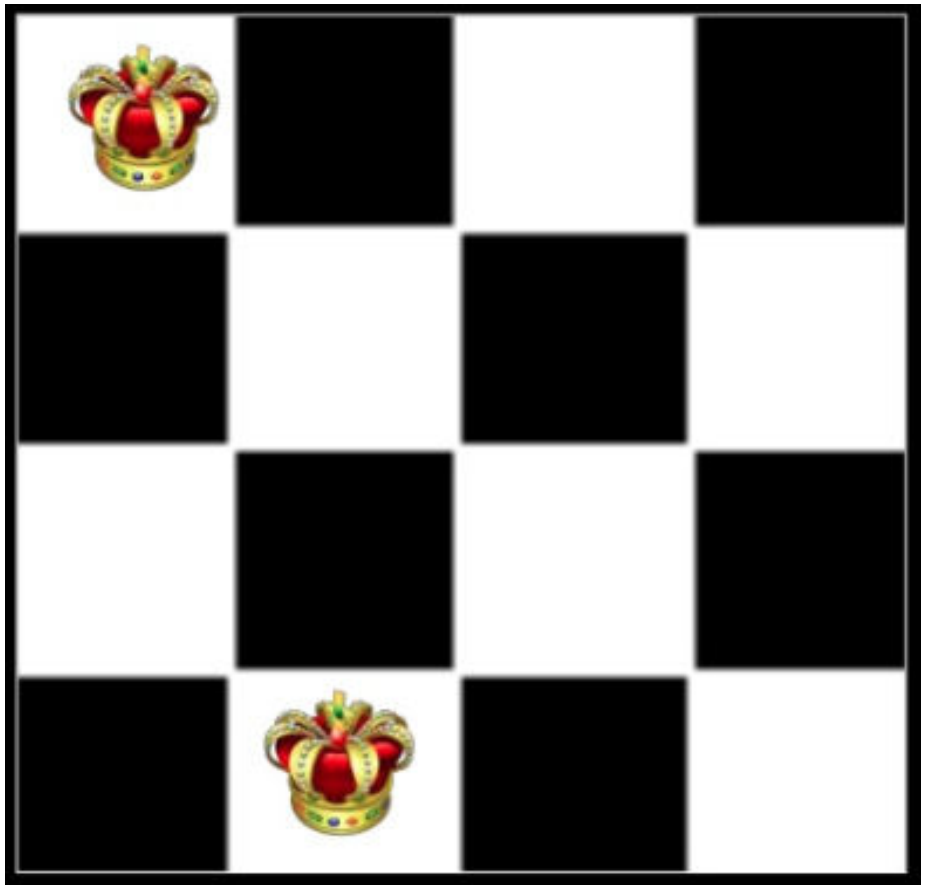
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



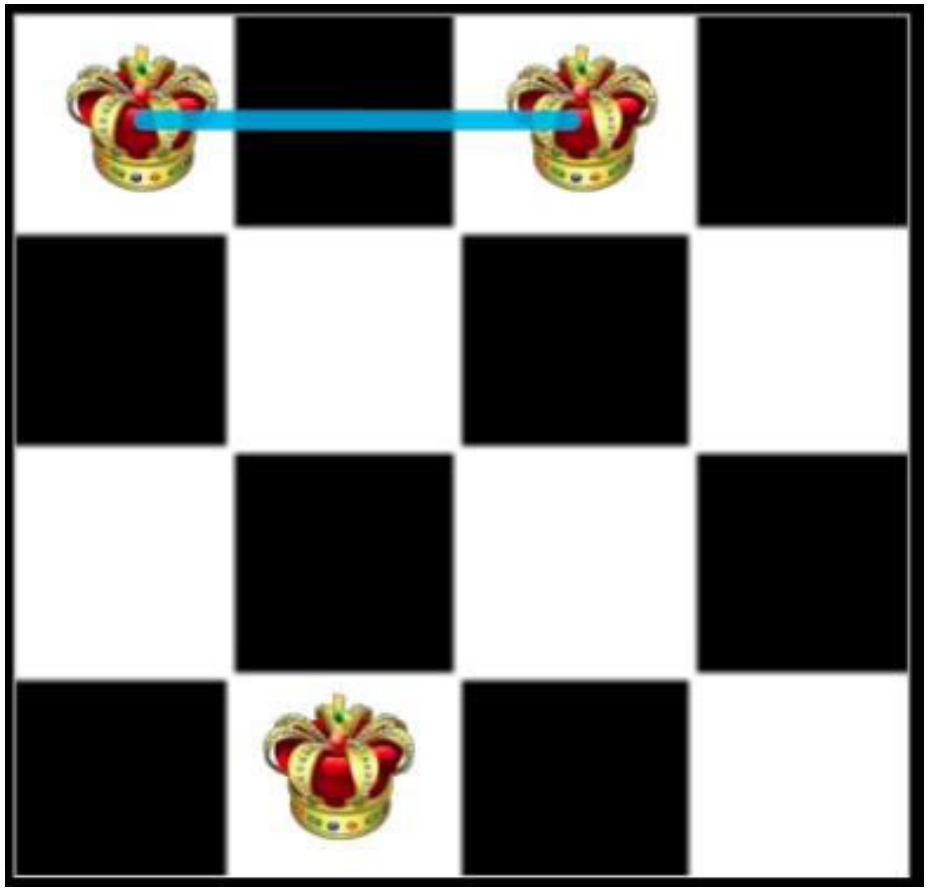
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



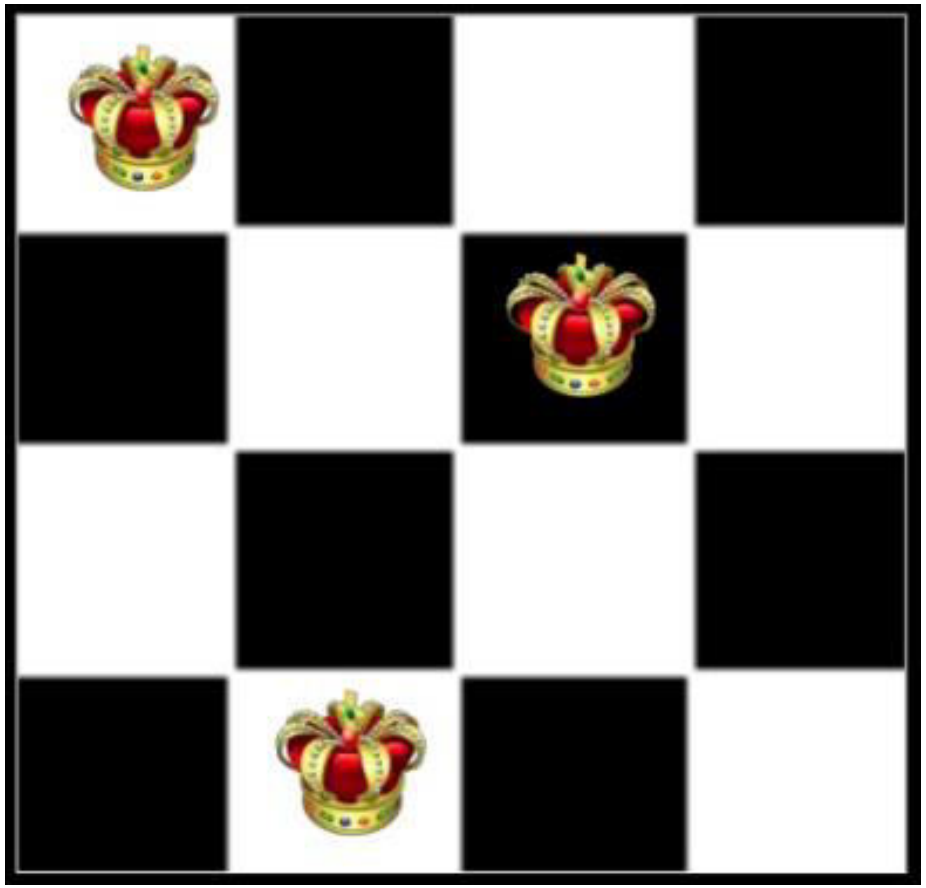
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



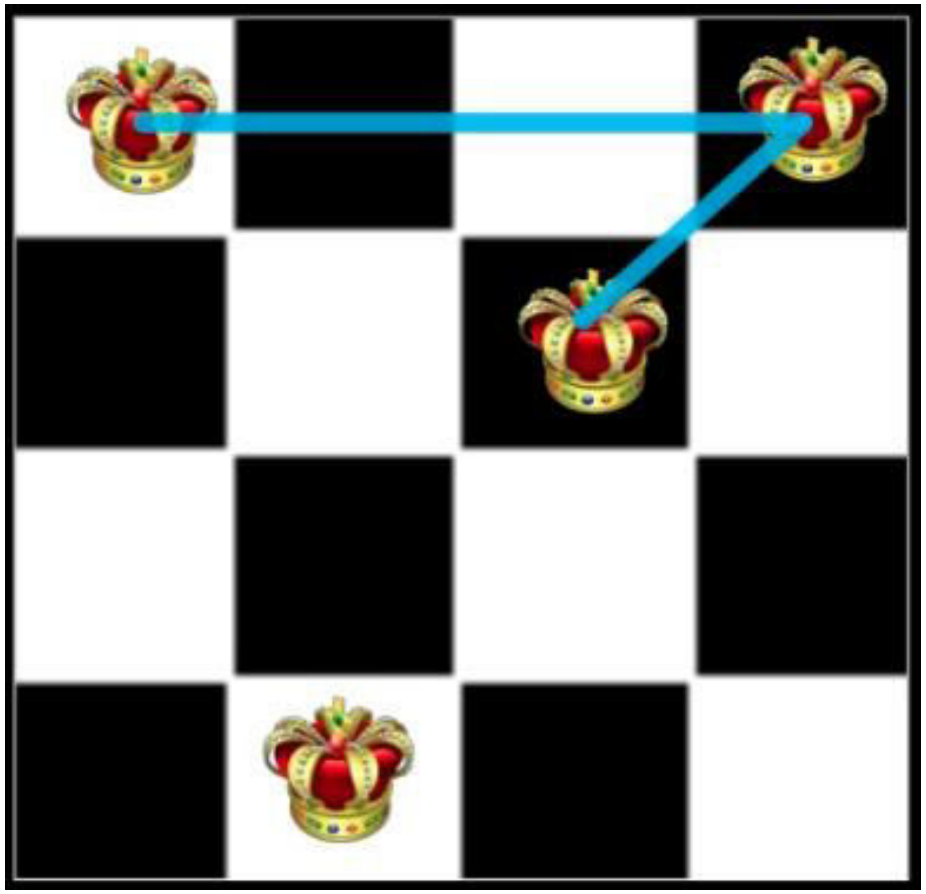
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



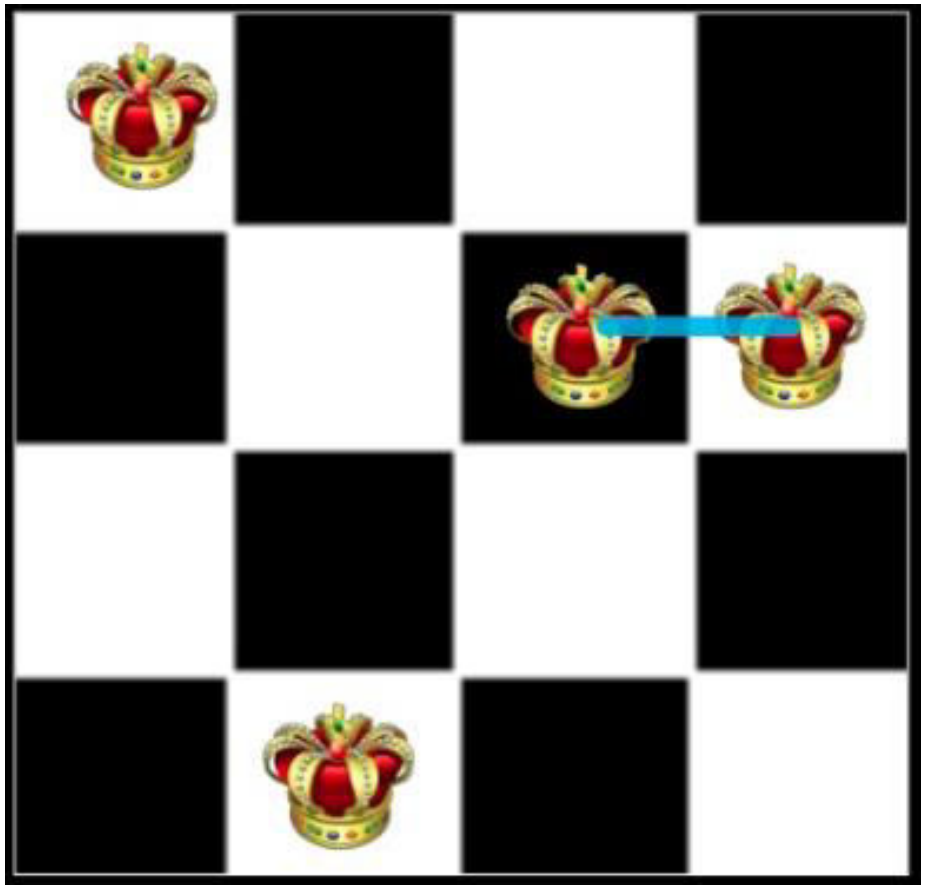
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



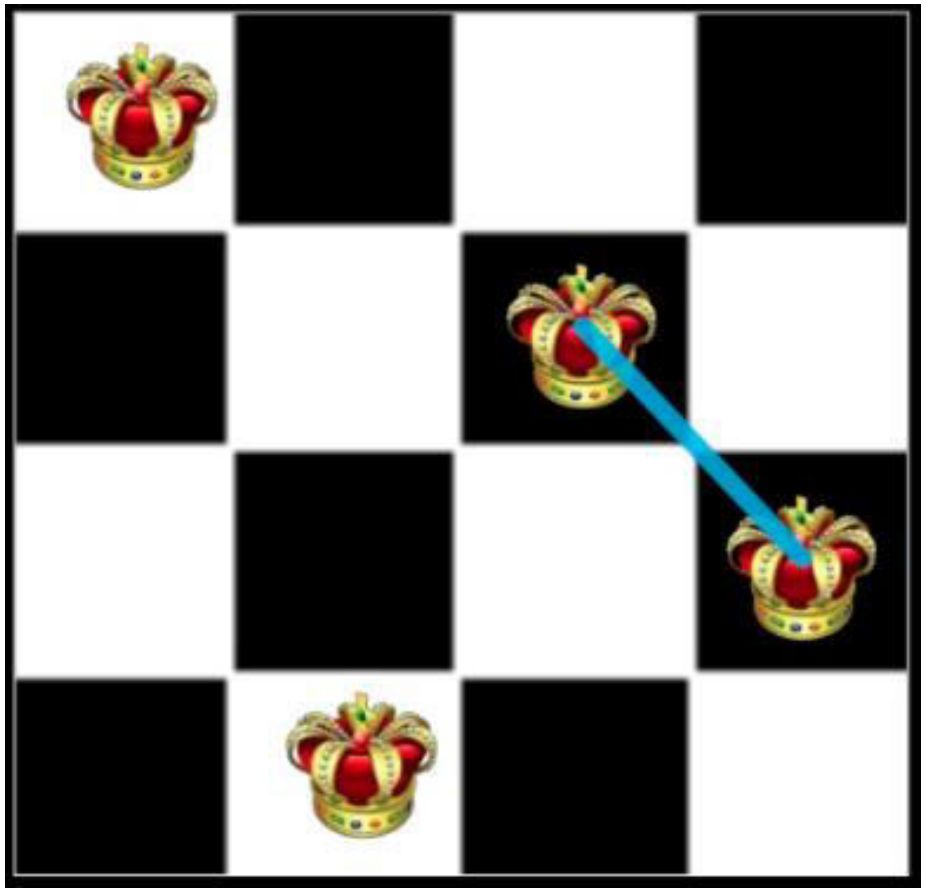
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



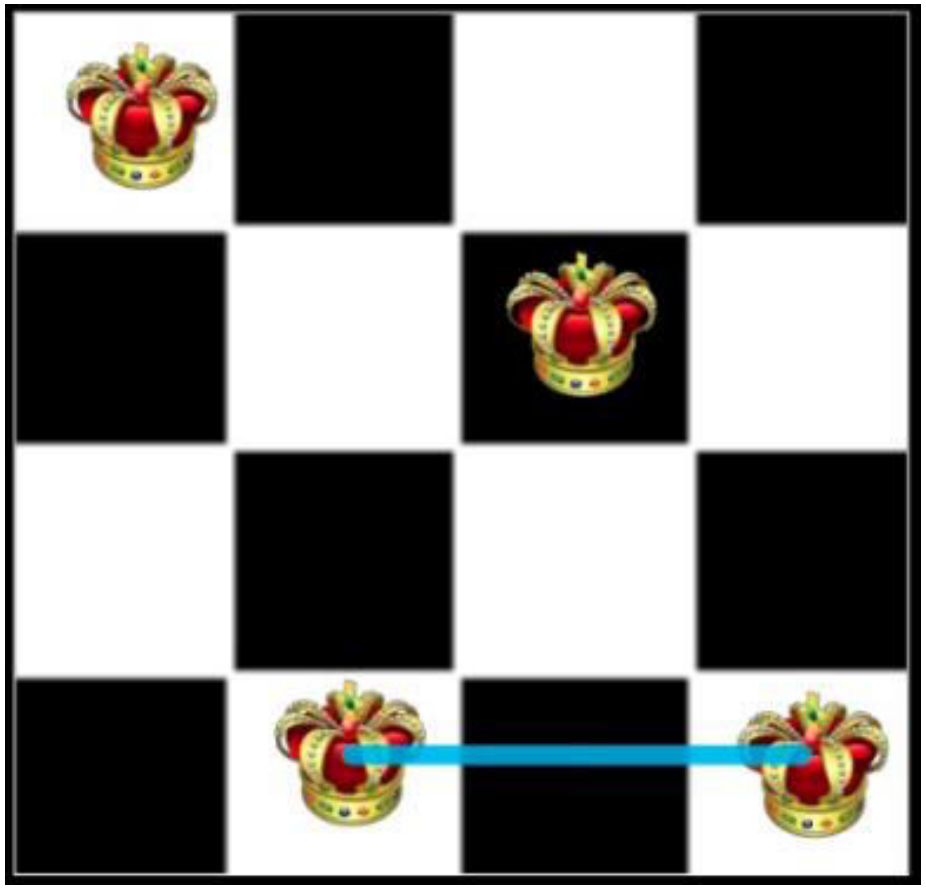
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



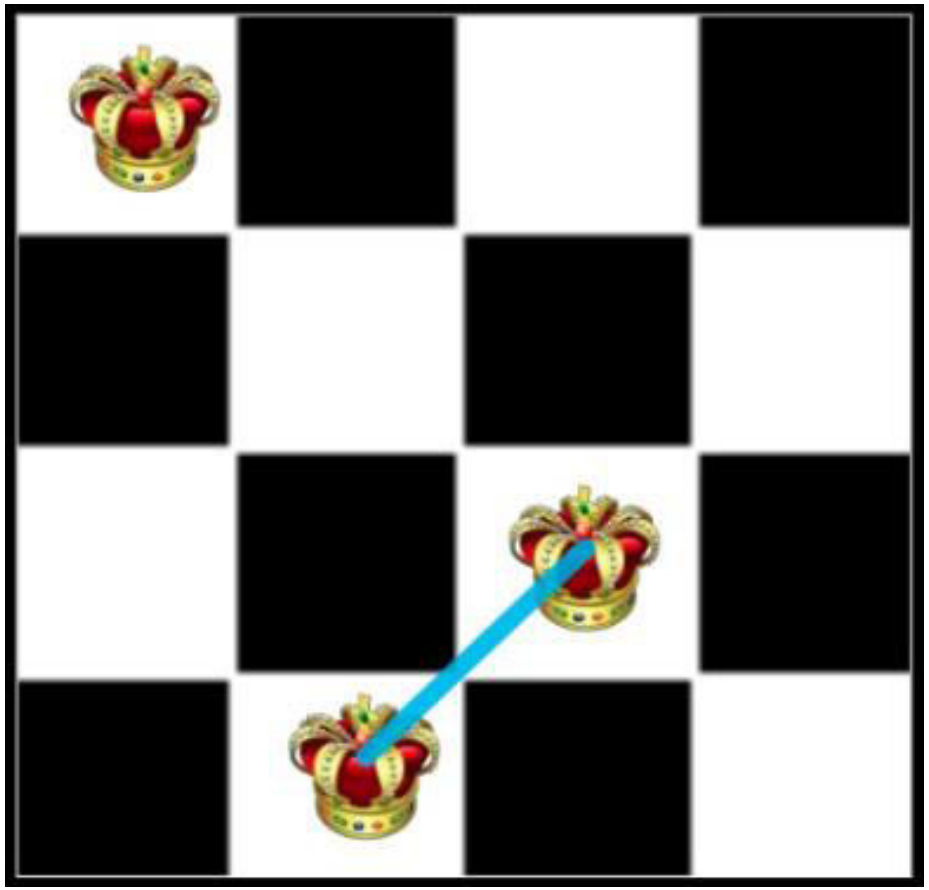
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



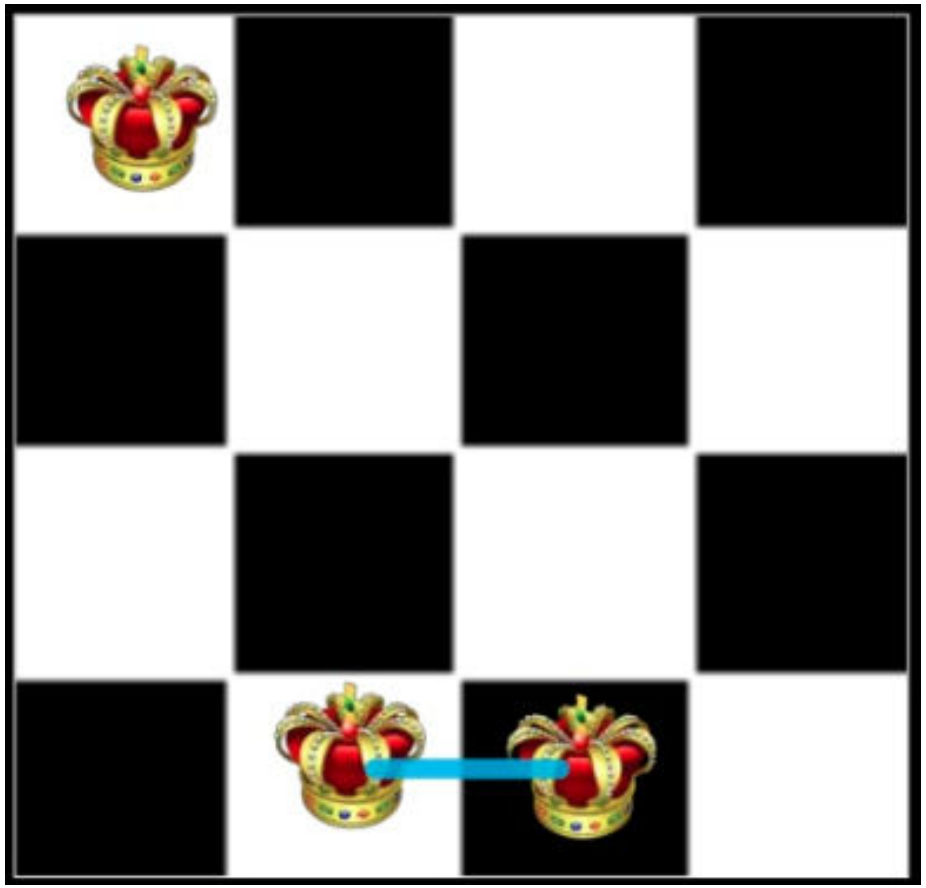
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



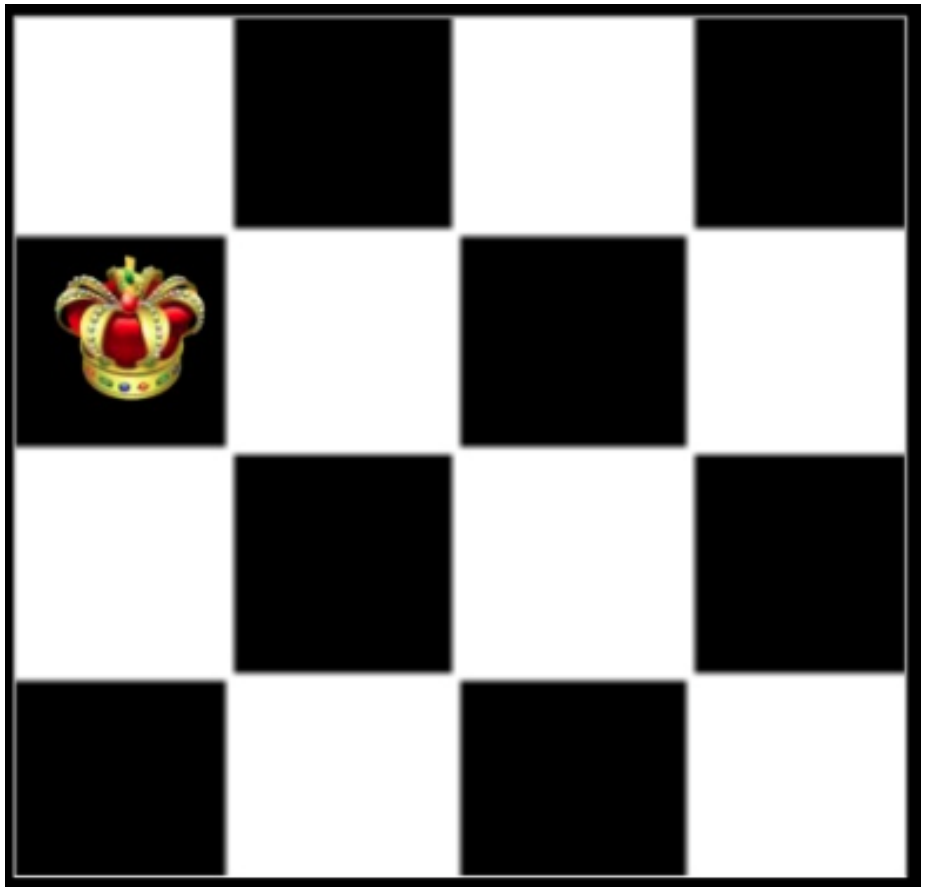
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



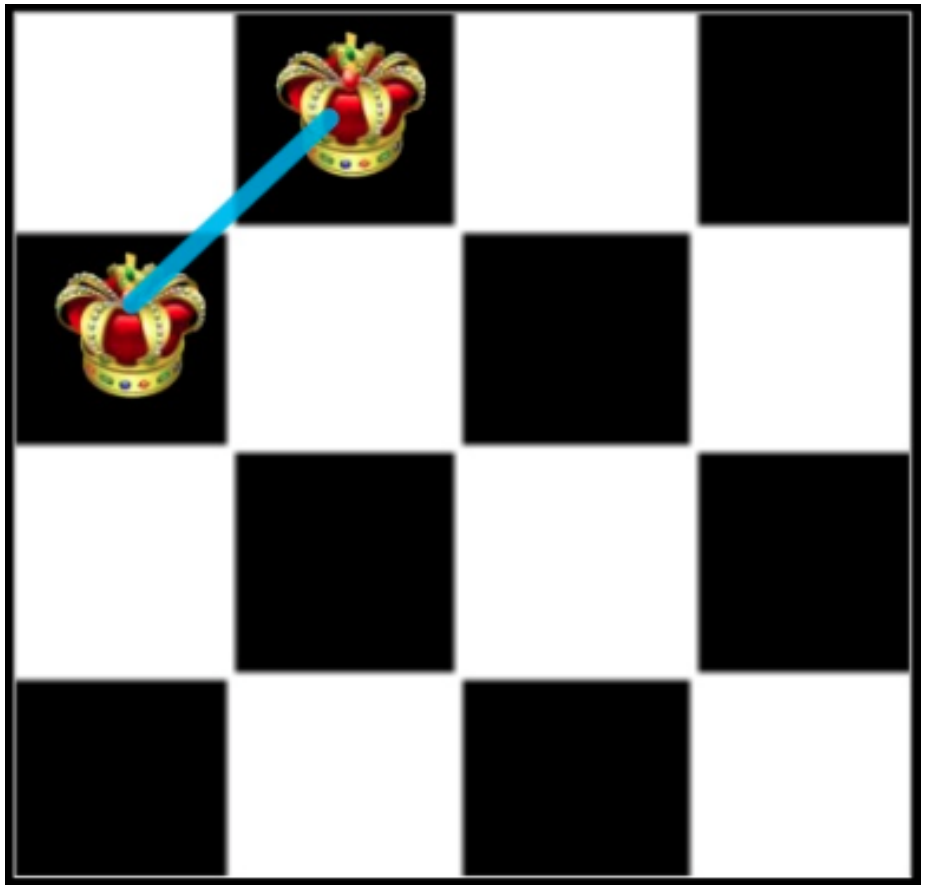
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



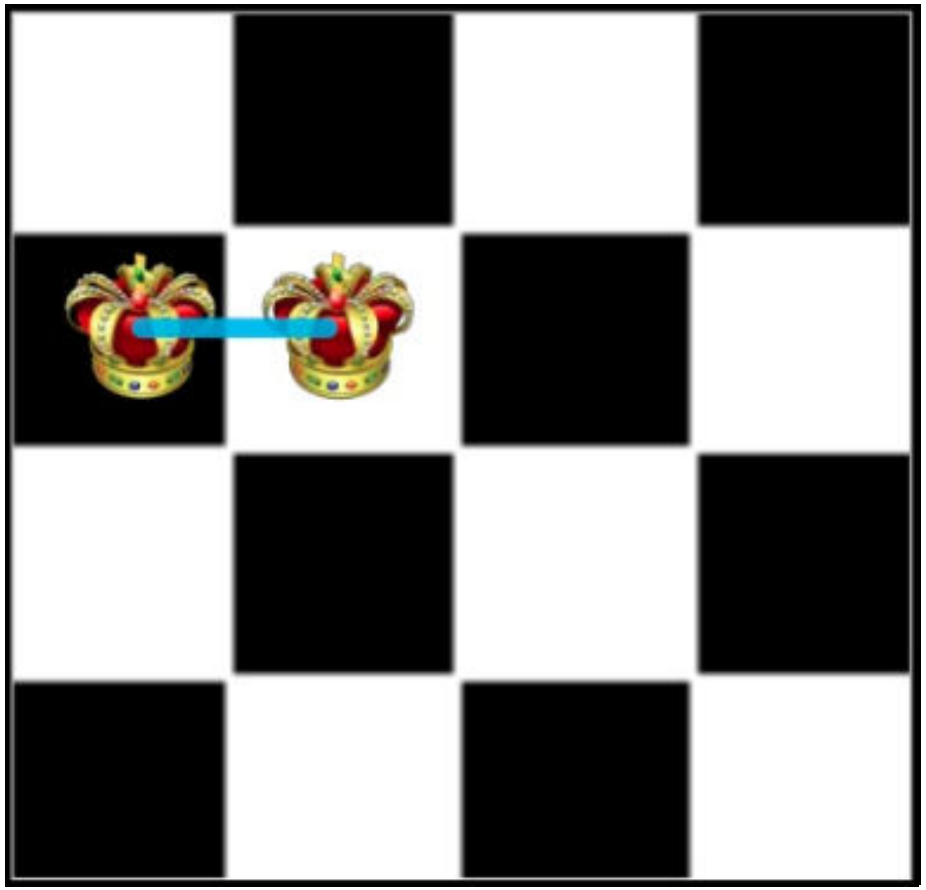
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



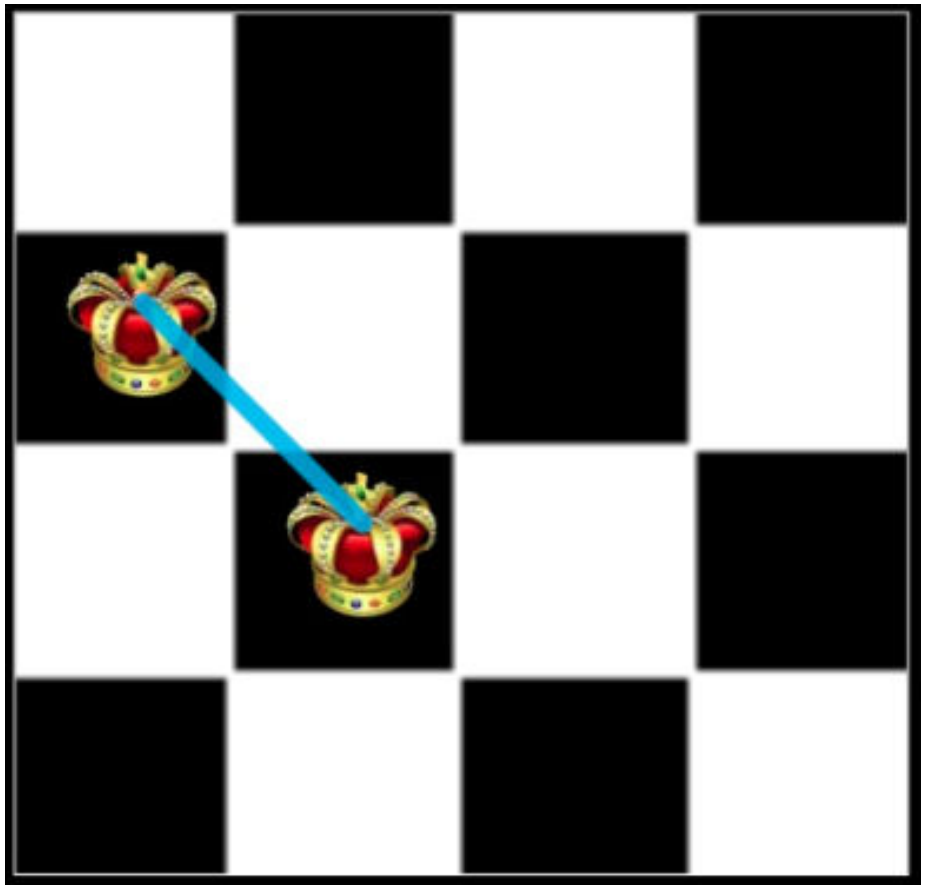
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



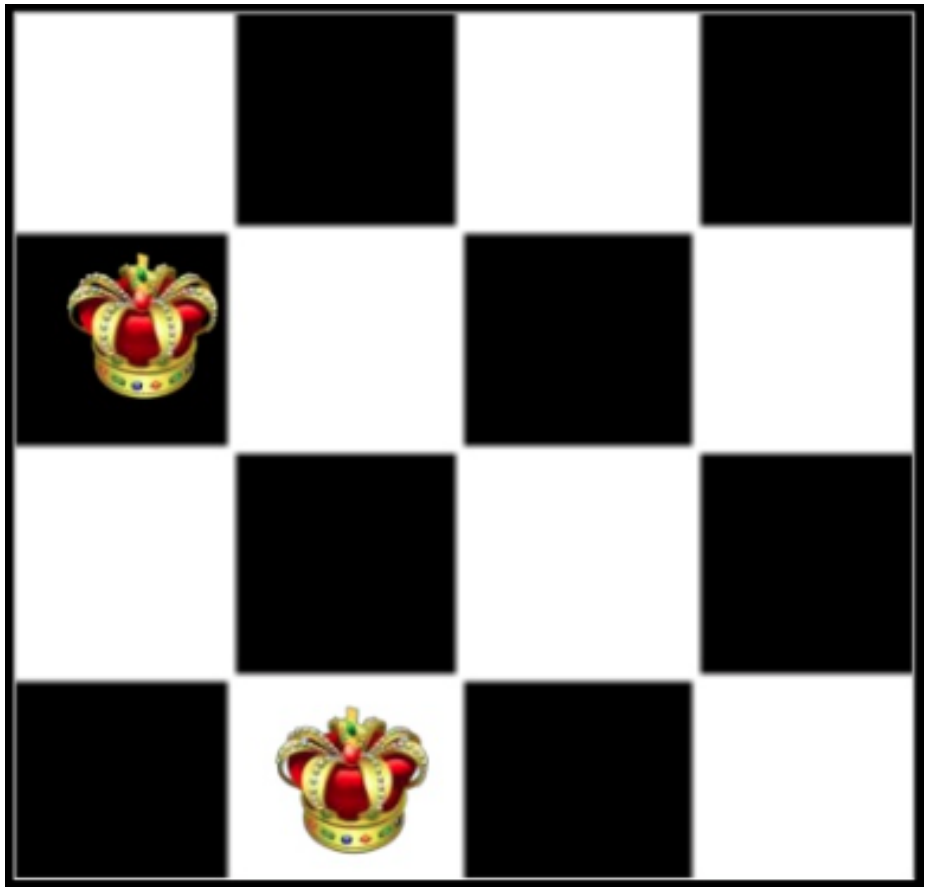
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



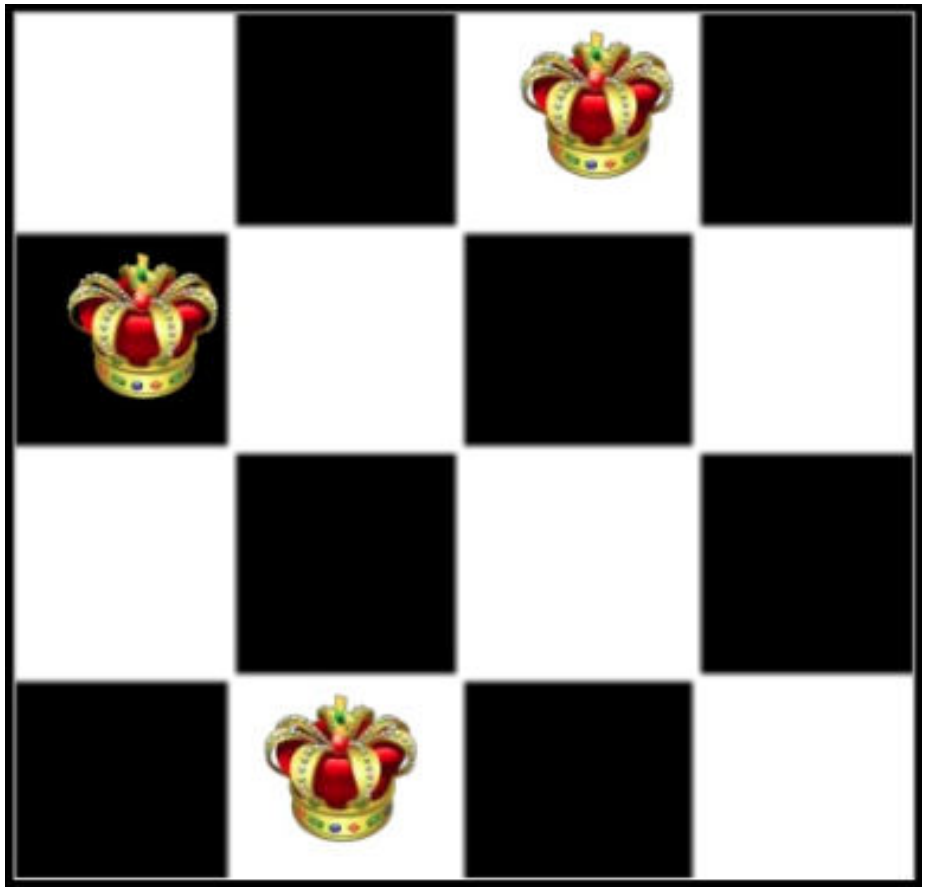
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



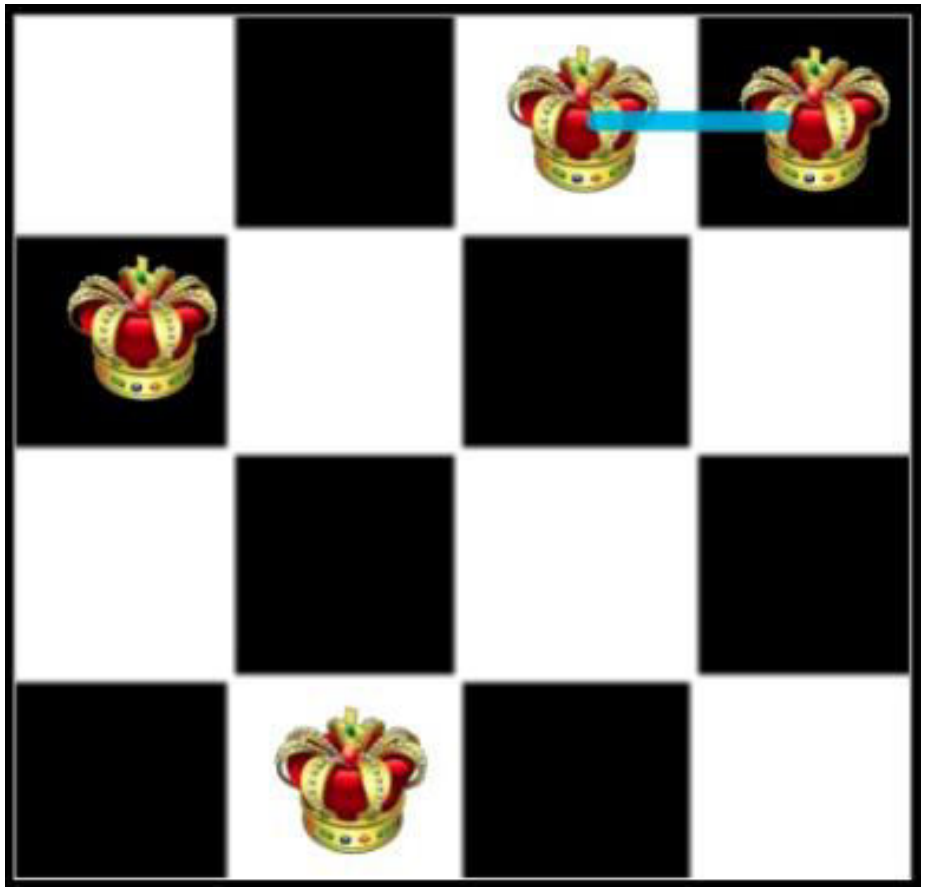
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



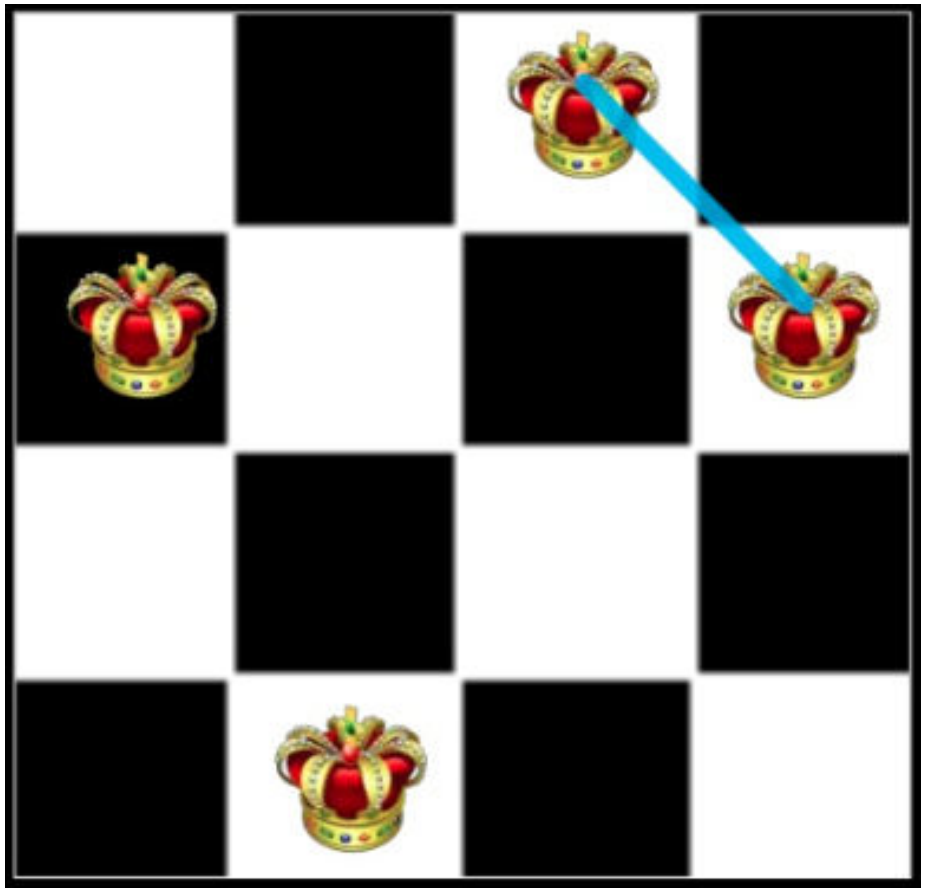
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



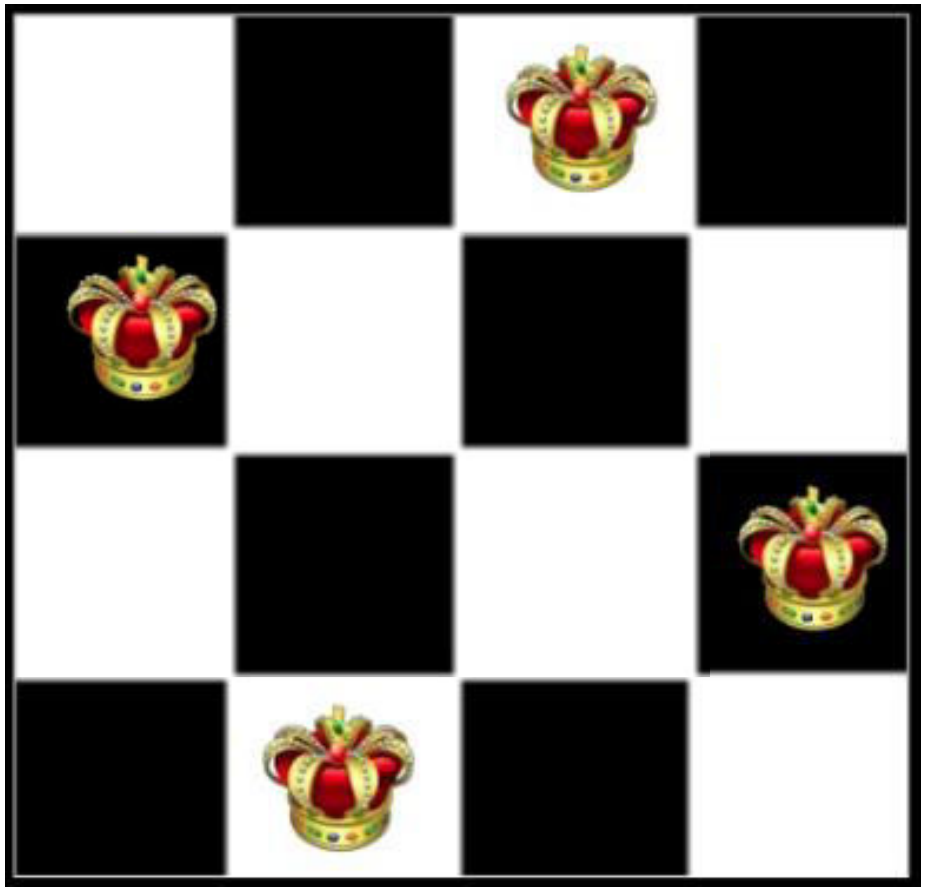
Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



Backtracking – problem n kraljica

- Ukoliko određena pozicija ne vodi rješenju problema, tj. ukoliko je k-ta kraljica napadnuta od strane neke od prethodno pozicioniranih kraljica, potrebno je vratiti se korak unazad i promijeniti poziciju kraljice k-1.



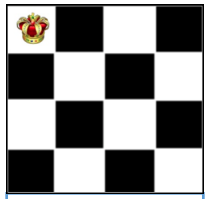
Backtracking – problem n kraljica

- Implementacija problema (pseudo kod):

```
void poredaj(int kraljica)
{
    if (kraljica == N) {
            prikaziRezultat();
    }
    else {
            for (i = 0 : N) {
                    if (nikoNeNapada(kraljica, i)){
                        postaviKraljicu(kraljica, i);
                        poredaj(kraljica+1);
                    }
            }
    }
}
```

Backtracking – problem n kraljica

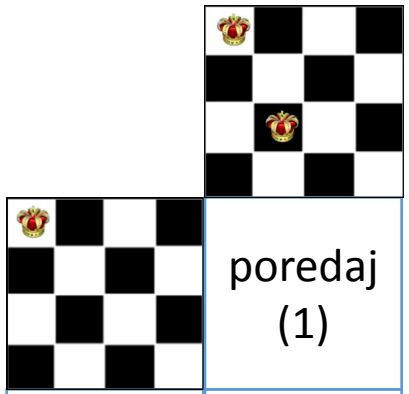
- Analiza rekurzivnih poziva



poredaj
(0)

Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



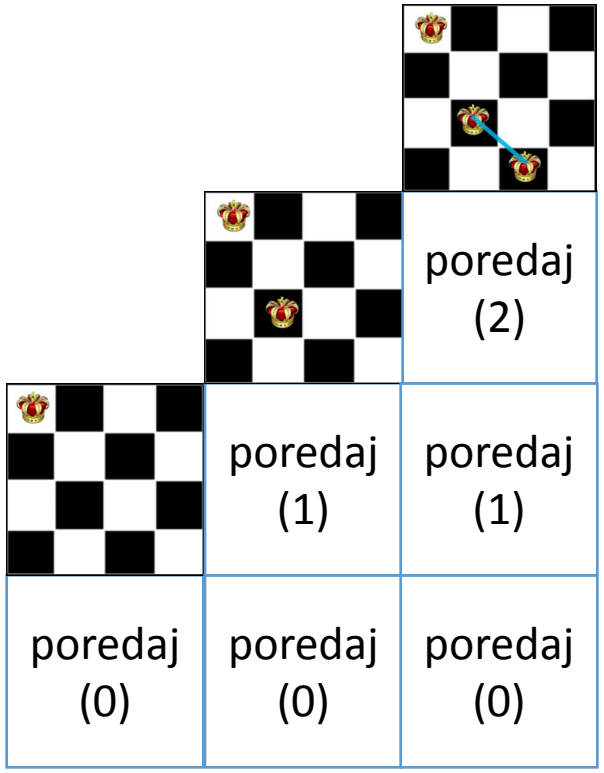
poredaj (1)

poredaj (0)

poredaj (0)

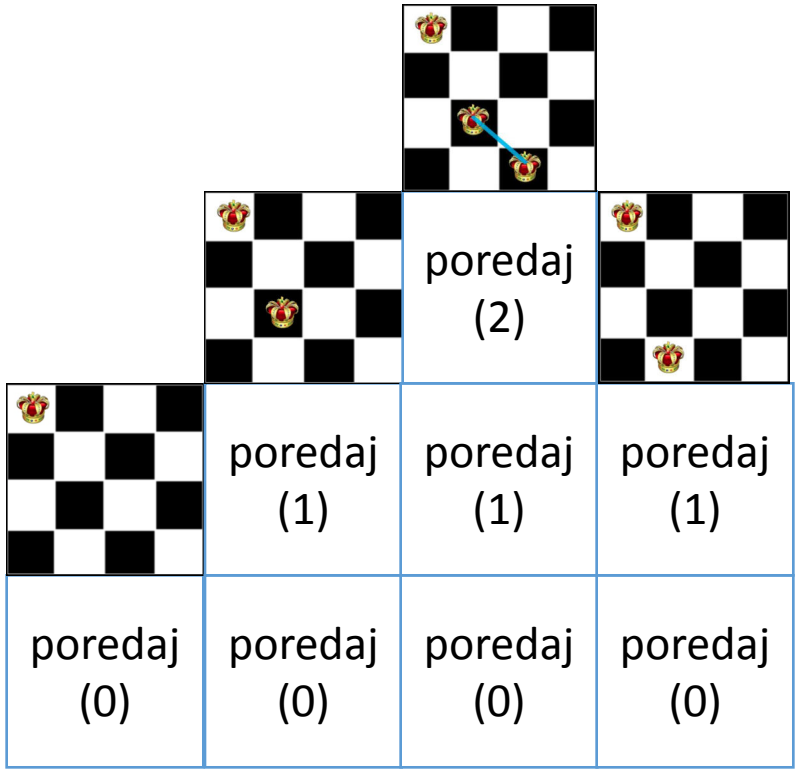
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



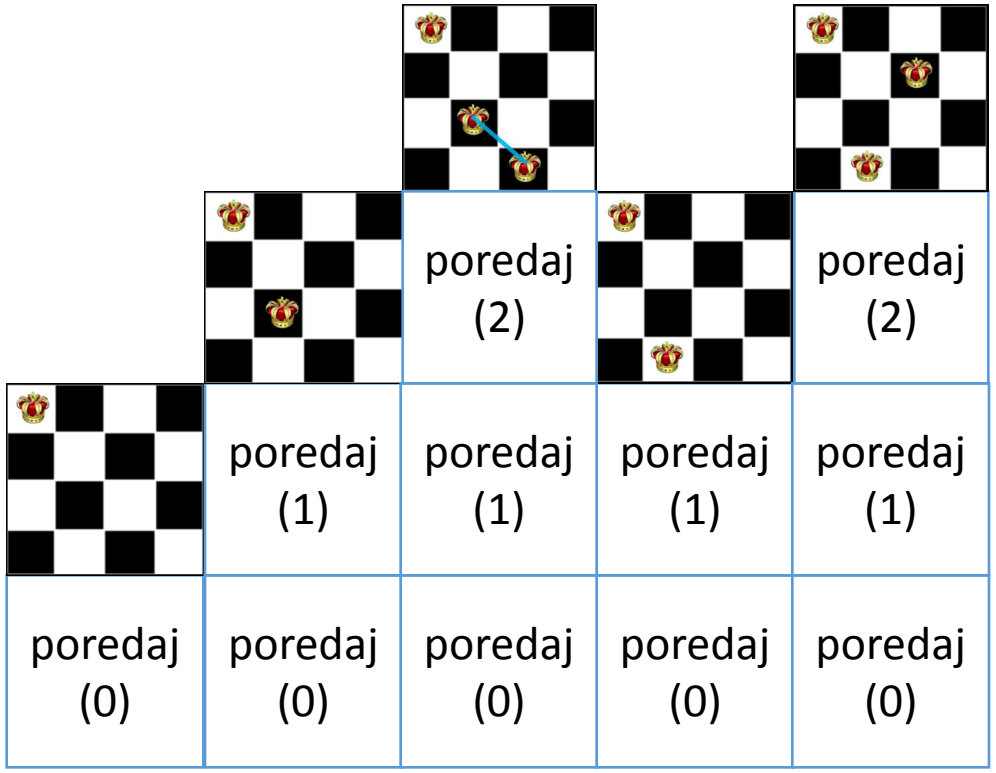
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



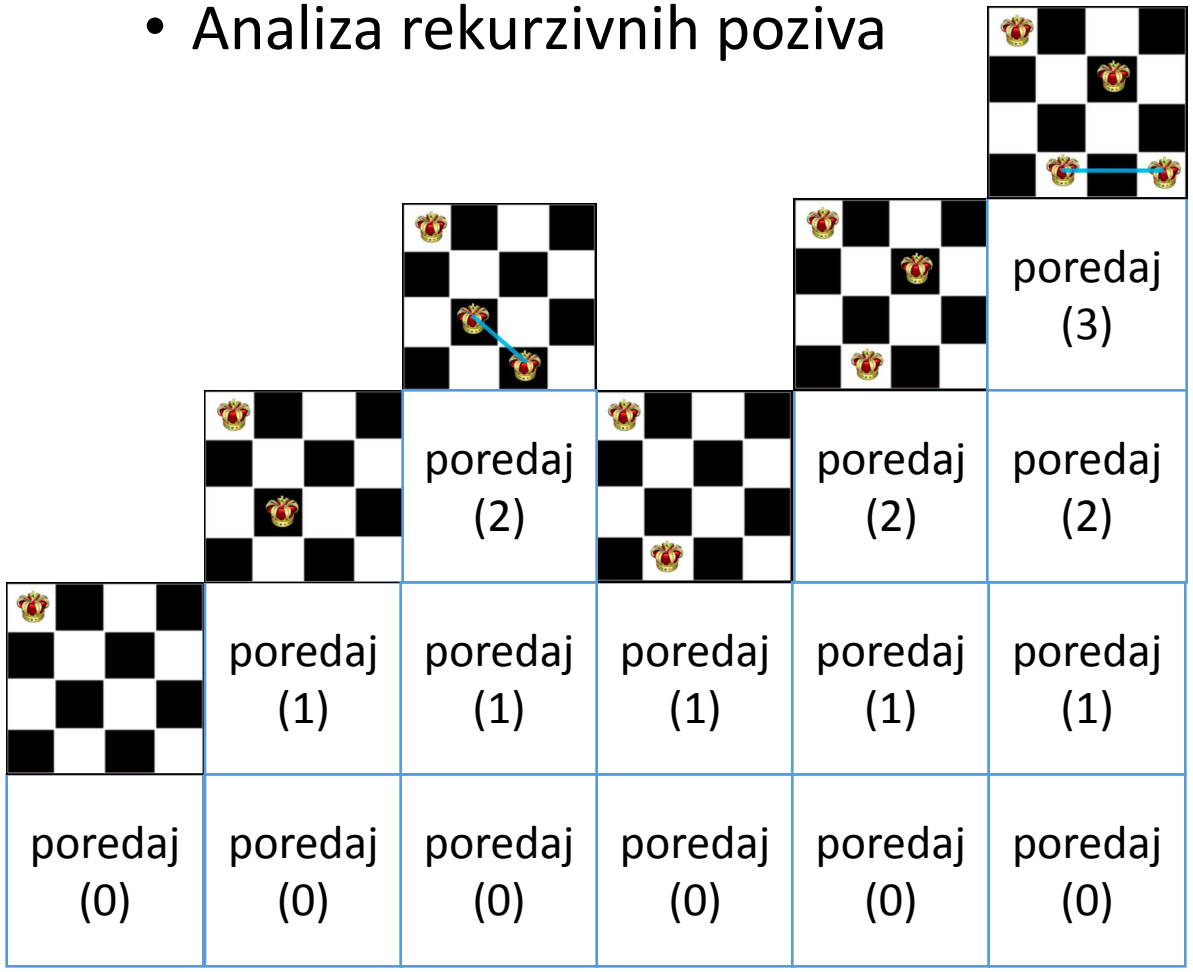
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



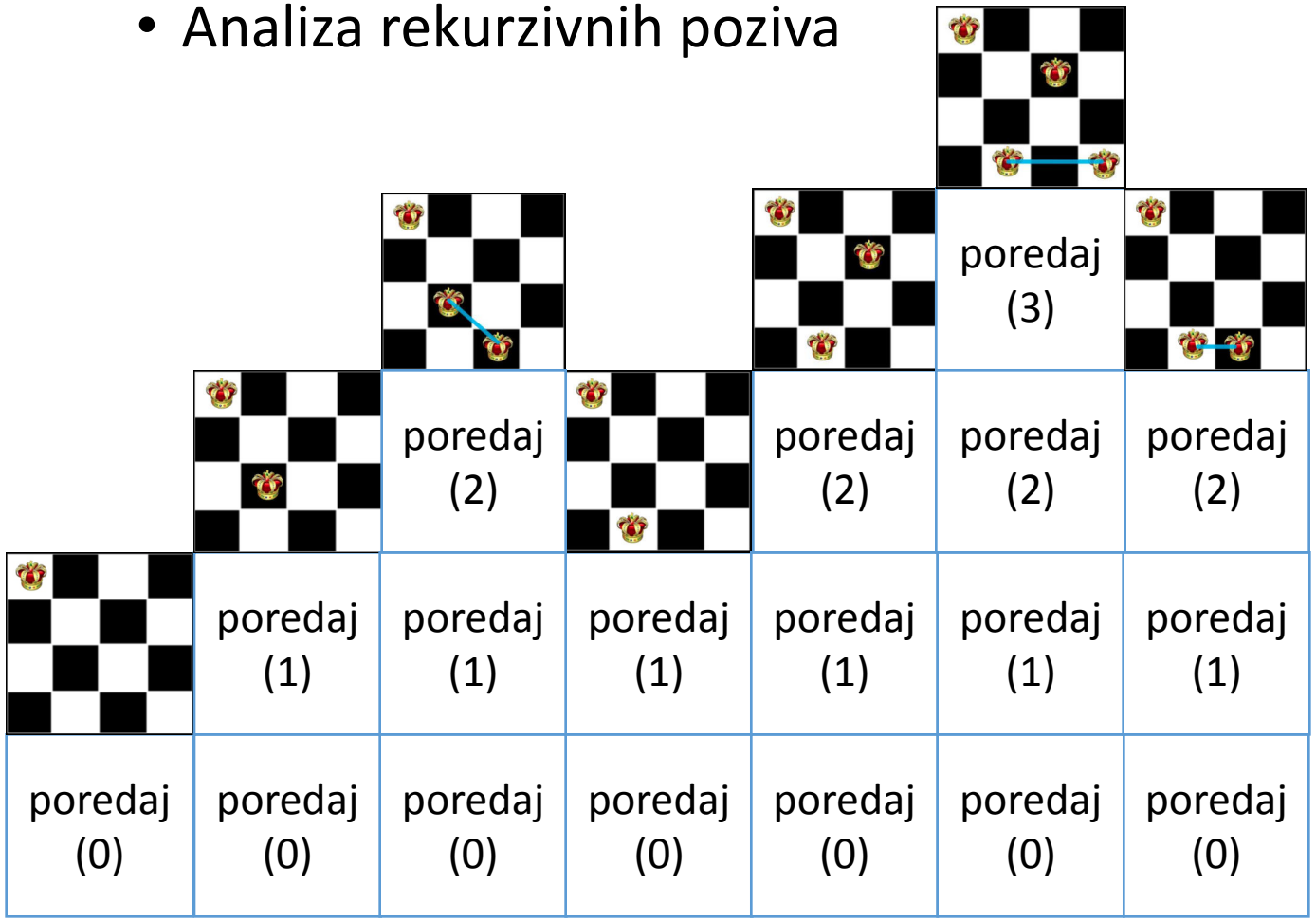
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



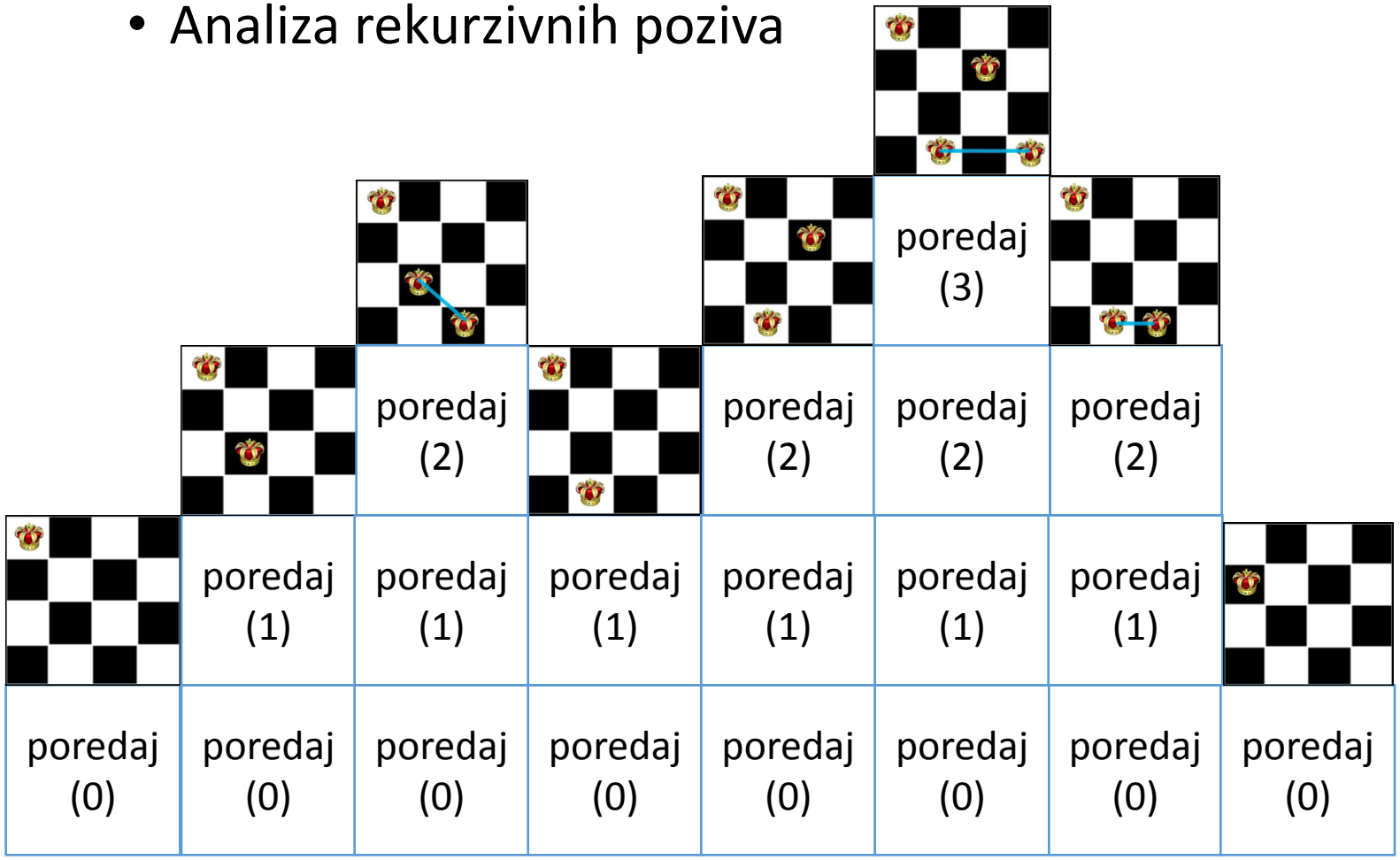
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



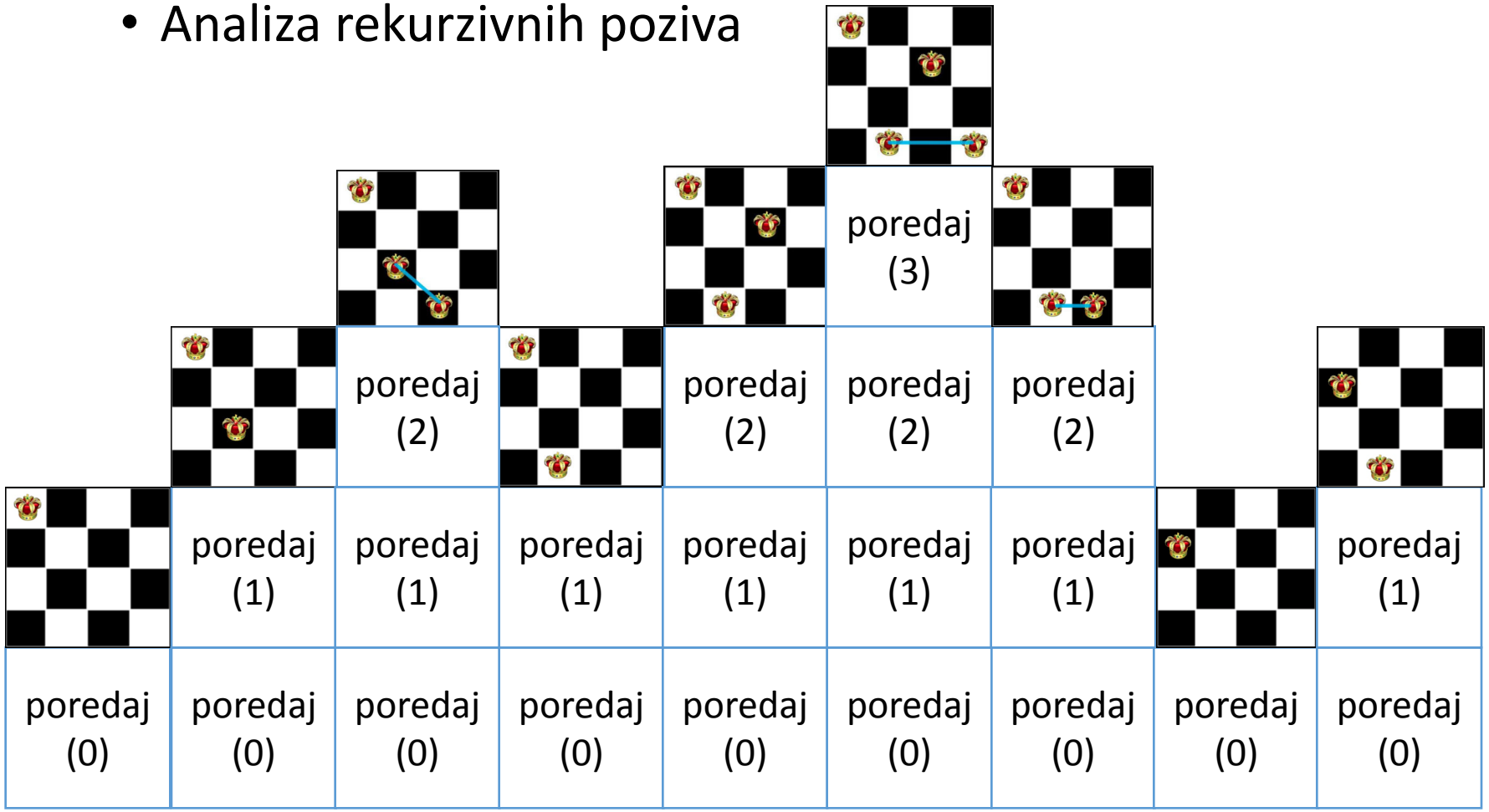
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



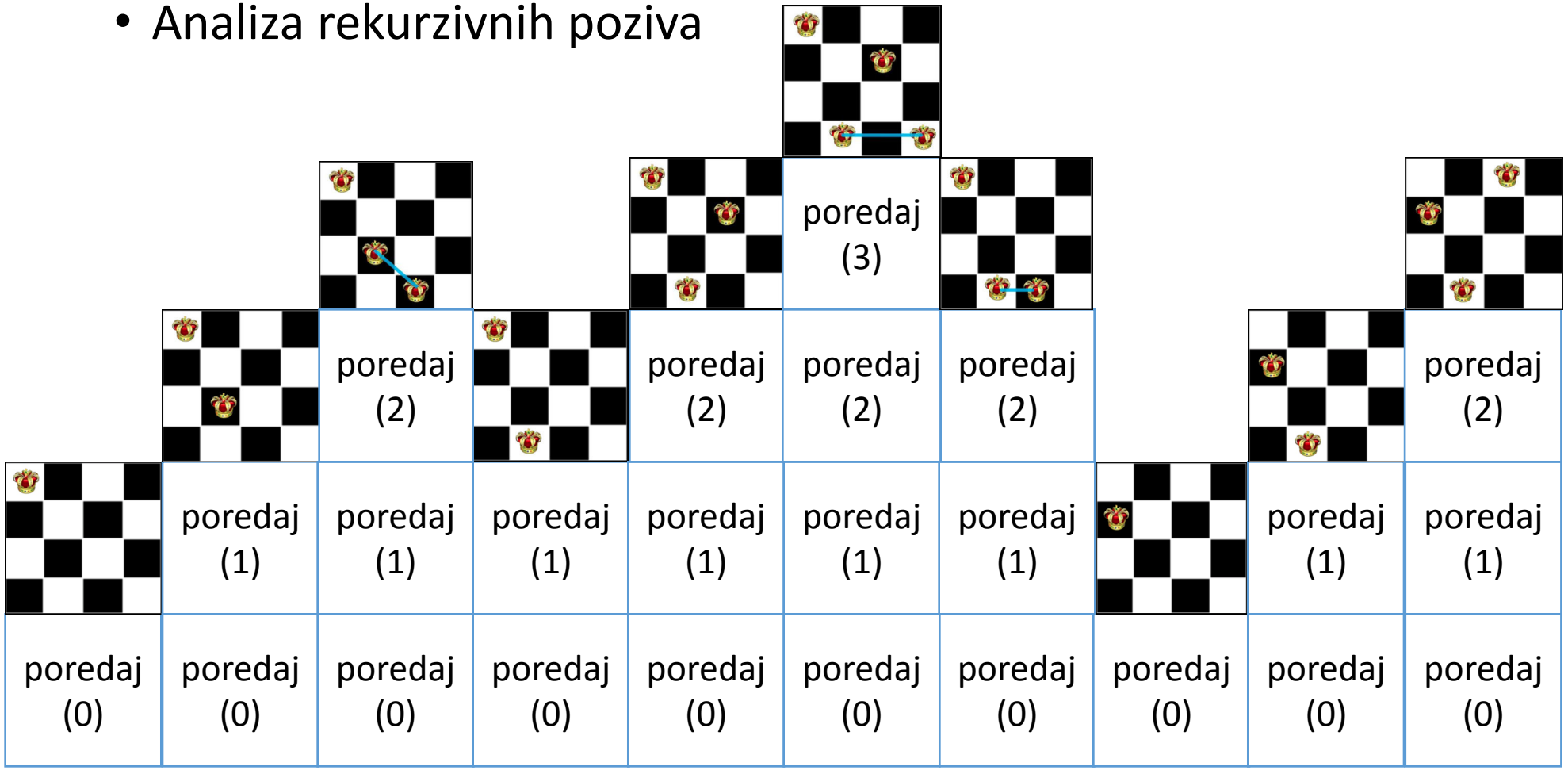
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



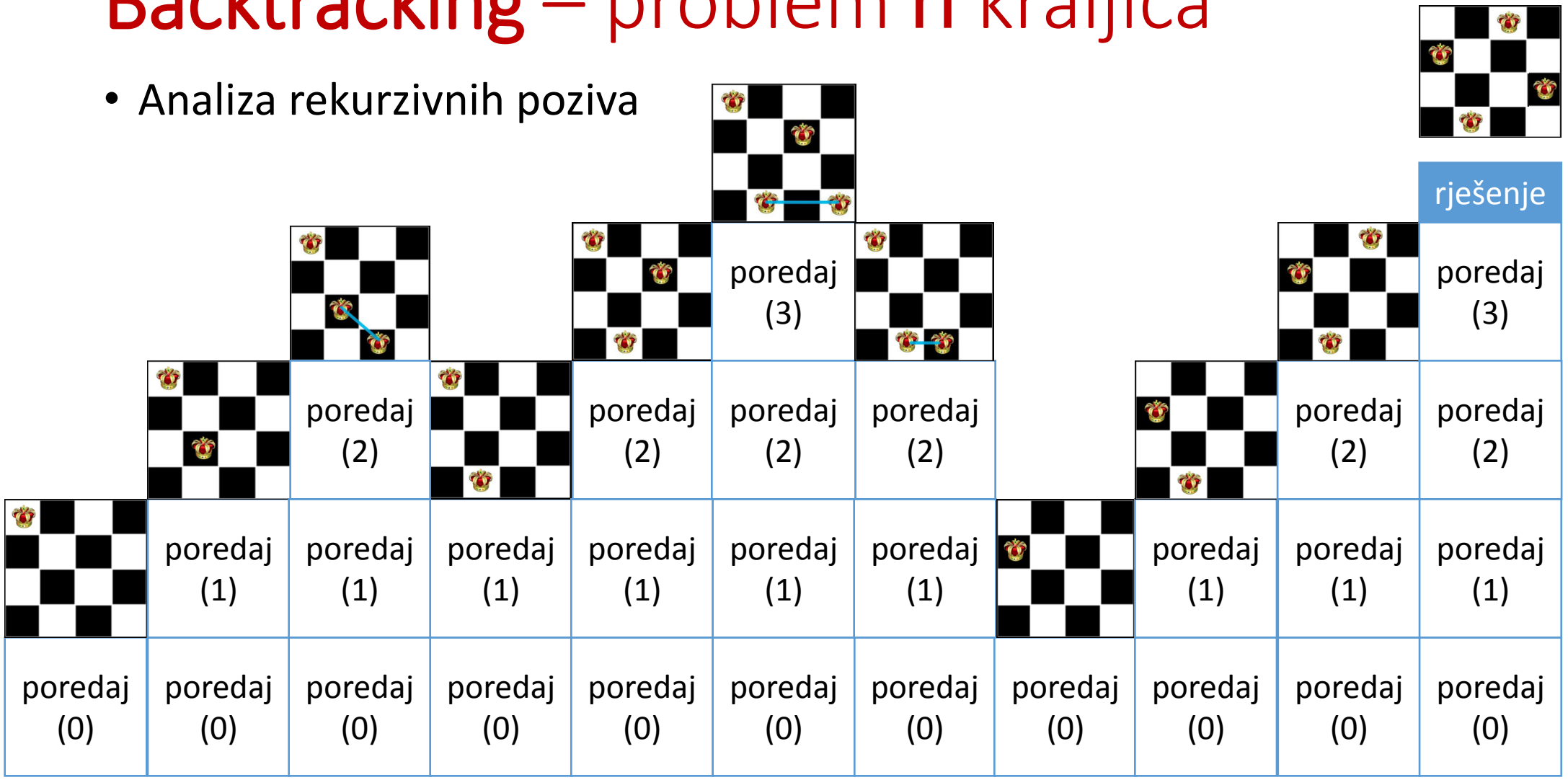
Backtracking – problem n kraljica

- Analiza rekurzivnih poziva



Backtracking – problem n kraljica

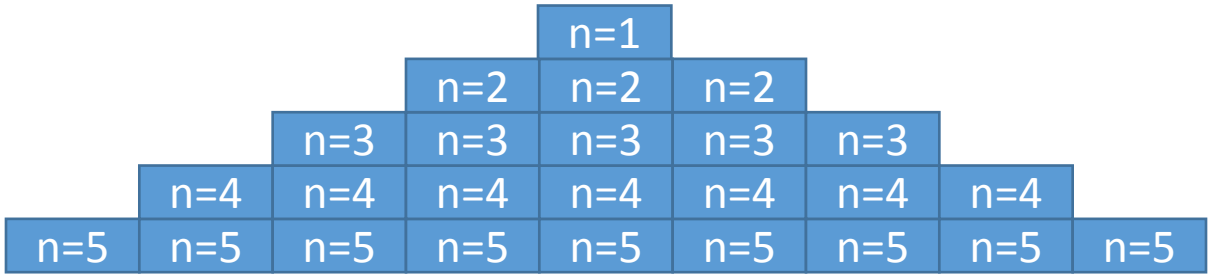
- Analiza rekurzivnih poziva



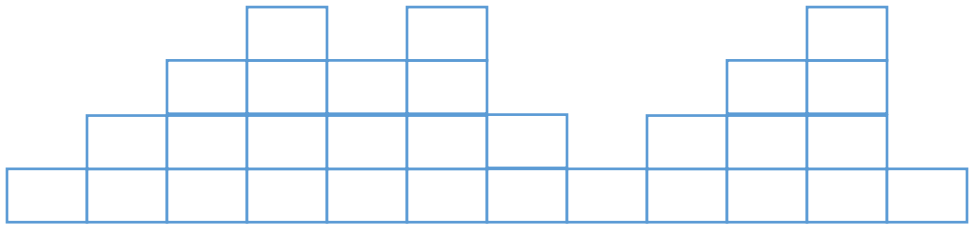
Backtracking

- Poređenje rekurzivnih *backtracking* poziva sa “običnim” rekurzijama

```
int faktorijel(int n){
    if(n==1)
            return 1;
    return n*faktorijel(n-1);
}
```



“obična” rekurzija



backtracking rekurzija