روش‌های جست‌وجوی فضای حالت

روش پس‌گرد (backtracking)

مسئله‌ی هشت‌وزیر

$\pi_i =$ سطر وزیر در ستون i. هدف پیداکردن جای‌گشتی از ۱ تا ۸ است که

$$\forall i, j \;\; |\pi_i - \pi_j| \neq |i - j|$$

راه‌حل کورکورانه

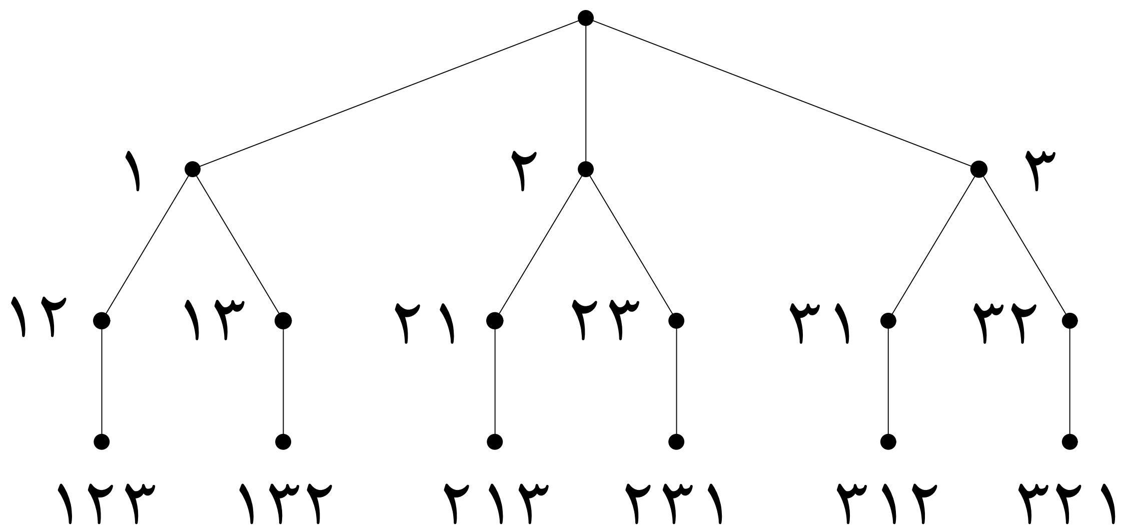
```
EightQueens()
  for i_1 ← 1 to 8
      do for i_2 ← 1 to 8
             do for i_3 ← 1 to 8
                    do ...
                        for i_8 ← 1 to 8
                            do try ← (i_1, i_2, ..., i_8)
                               if Solution (try)
                                 then Print try
```

تولید فضای حالت

Permutation(A, i)

$\triangleright$ Initilization: $A[i] = i$, $1 \leq i \leq n$
$\triangleright$ assuming that $A[1..i-1]$ has been selected
$\triangleright$ find all permutations of $A[1..n]$

$n \leftarrow$ TT Length (A)
if $i > n$
    then Print $A[1..n]$
    else **for** $j \leftarrow i$ **to** n
        do Swap $(A[i], A[j])$
            Permutation $(A, i+1)$
            Swap $(A[i], A[j])$



۱
۲
۳
۱۲
۱۳
۲۱
۲۳
۳۱
۳۲
۱۲۳
۱۳۲
۲۱۳
۲۳۱
۳۱۲
۳۲۱

```
N-QUEEN(A, i)
      ▷ print the permutation that is feasible
n ← LENGTH (A)
if i > n
   then for i ← 1 to n
              do for j ← i + 1 to n
                     do if |A[i] − A[j]| = |i − j|
                           then exit loop
           PRINT A[1..n]
   else  for j ← i to n
              do SWAP (A[i], A[j])
                 N-QUEEN (A, i + 1)
                 SWAP (A[i], A[j])
```

```
FEASIBLE(A, i)
    ▷ Assuming that A[1..i − 1] is feasible,
    ▷ it checks whether A[1..i] is feasible
if i > LENGTH(A)
   then return true
for j ← 1 to i − 1
    do if |A[i] − A[j]| = |i − j|
          then return false
return true
```

```
N-QUEEN(A, i)
      ▷ print the permutation that is feasible
n ← LENGTH (A)
if i > n
   then PRINT A[1..n]
   else  for j ← i to n
              do SWAP (A[i], A[j])
                 if not FEASIBLE(i)
                    then return
                 N-QUEEN (A, i + 1)
                 SWAP (A[i], A[j])
```

بردار k-promising

N-Queens$(k, col, diag45, diag135)$
$\triangleright$ try is k-promising array $[1..k]$
$\triangleright$ $col \leftarrow \{try[i] : 1 \leq i \leq k\}$
$\triangleright$ $diag45 \leftarrow \{try[i] - i + 1 : 1 \leq i \leq k\}$
$\triangleright$ $diag135 \leftarrow \{try[i] + i - 1 : 1 \leq i \leq k\}$
if $k = N$ {an N-promising verstor is a solution}
  then Print try
  else { find $(k+1)$-promising extensioun}
    for $j \leftarrow 1$ **to** N
      do if $j \notin col$ **and** $j - k \notin diag45$ **and** $j + k \notin diag135$
        then $try[k+1] \leftarrow j$
        N-Queens$(k+1, col + \{j\}, diag45 + \{j-k\}, diag135 + \{j+k\})$

توجه

برای ۱۲ $= n$، ۴۷۹/۰۰۱/۶۰۰ جای‌گشت وجود دارد. اولین جواب در ۴/۵۴۶/۰۴۴ امین حلقه به‌دست می‌آید. ولی درخت حالت در روش پس‌گرد ۸۵۶/۱۸۹ گره دارد که در ۲۶۲ امین گره یک جواب به‌دست می‌آید.

ترتیب قرار دادن مدارها در رَک

بازسازی جاده‌ی کمربندی (Turnpike Reconstruction)

n نقطه‌ی p_1 تا p_n برروی محور x ها که $x_1 = 0$ داده شده‌اند. با $O(n^2)$ می‌توان همه‌ی $n(n-1)/2$ فاصله را پیدا کرد.

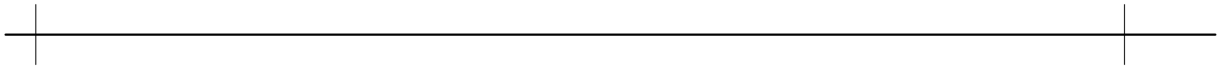
اگر مجموعه‌ی D شامل $n(n-1)/2$ فاصله داده شده باشند، آیا می‌توان نقاط را به‌دست آورد؟

مشخص نیست که آیا برای این مسئله راه‌حل چندجمله‌ای وجود دارد و آیا این مسئله ان‌پی-تمام است.

مثال

$$D = \{۱, ۲, ۲, ۲, ۳, ۳, ۳, ۴, ۵, ۵, ۵, ۶, ۷, ۸, ۱۰\}$$

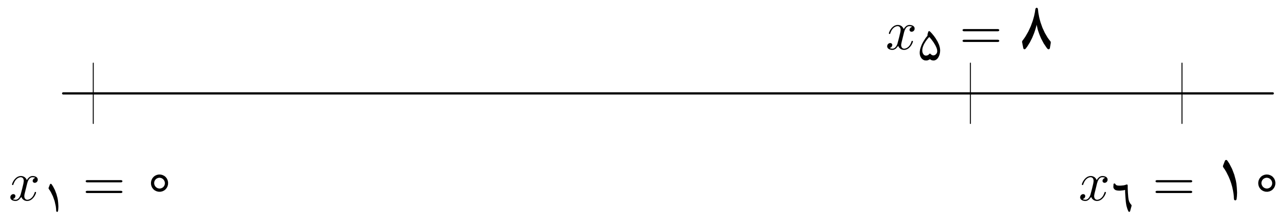
پس $n = ۶$



$x_1 = 0$ $\qquad\qquad$ $x_6 = 10$

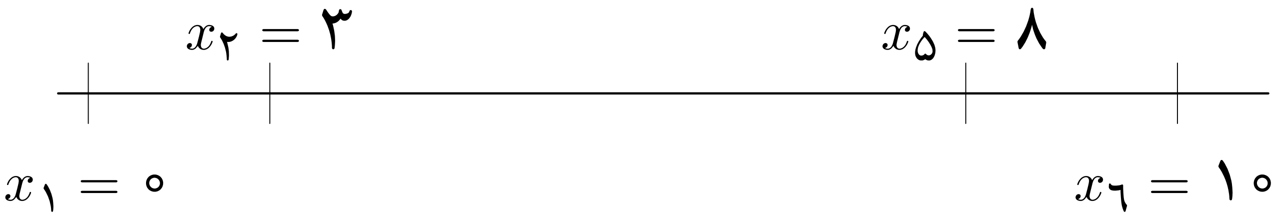
$$D = \{1, 2, 2, 2, 3, 3, 3, 4, 5, 5, 5, 6, 7, 8\}$$

پس از انتخاب x_1 و x_6.



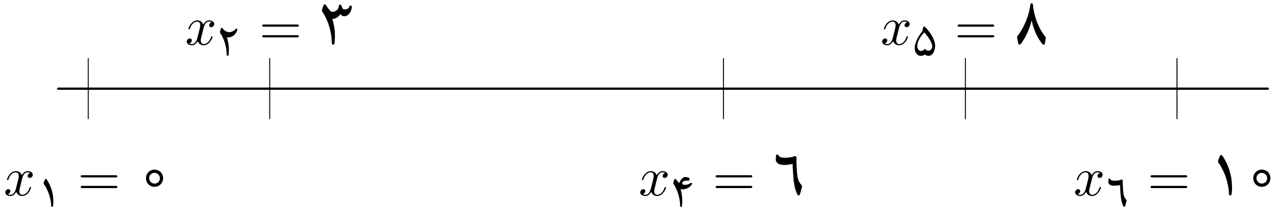
$$D = \{۱, ۲, ۲, ۳, ۳, ۳, ۴, ۵, ۵, ۵, ۶, ۷\}$$

پس از انتخاب $x_۱$، $x_۶$ و $x_۵$.



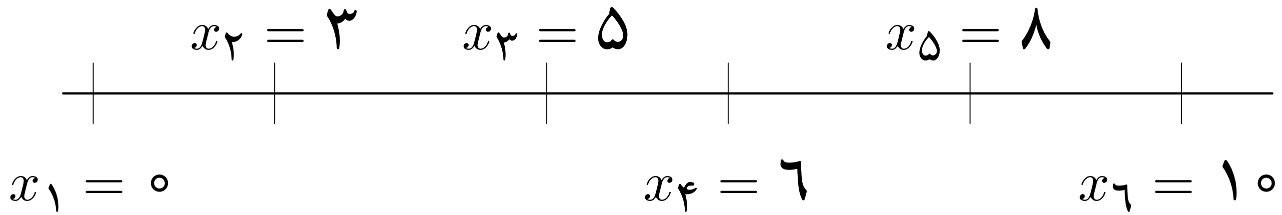
$$D = \{۱, ۲, ۲, ۳, ۳, ۴, ۵, ۵, ۶\}$$

پس از انتخاب $x_۱$، $x_۶$، $x_۵$ و $x_۲$.



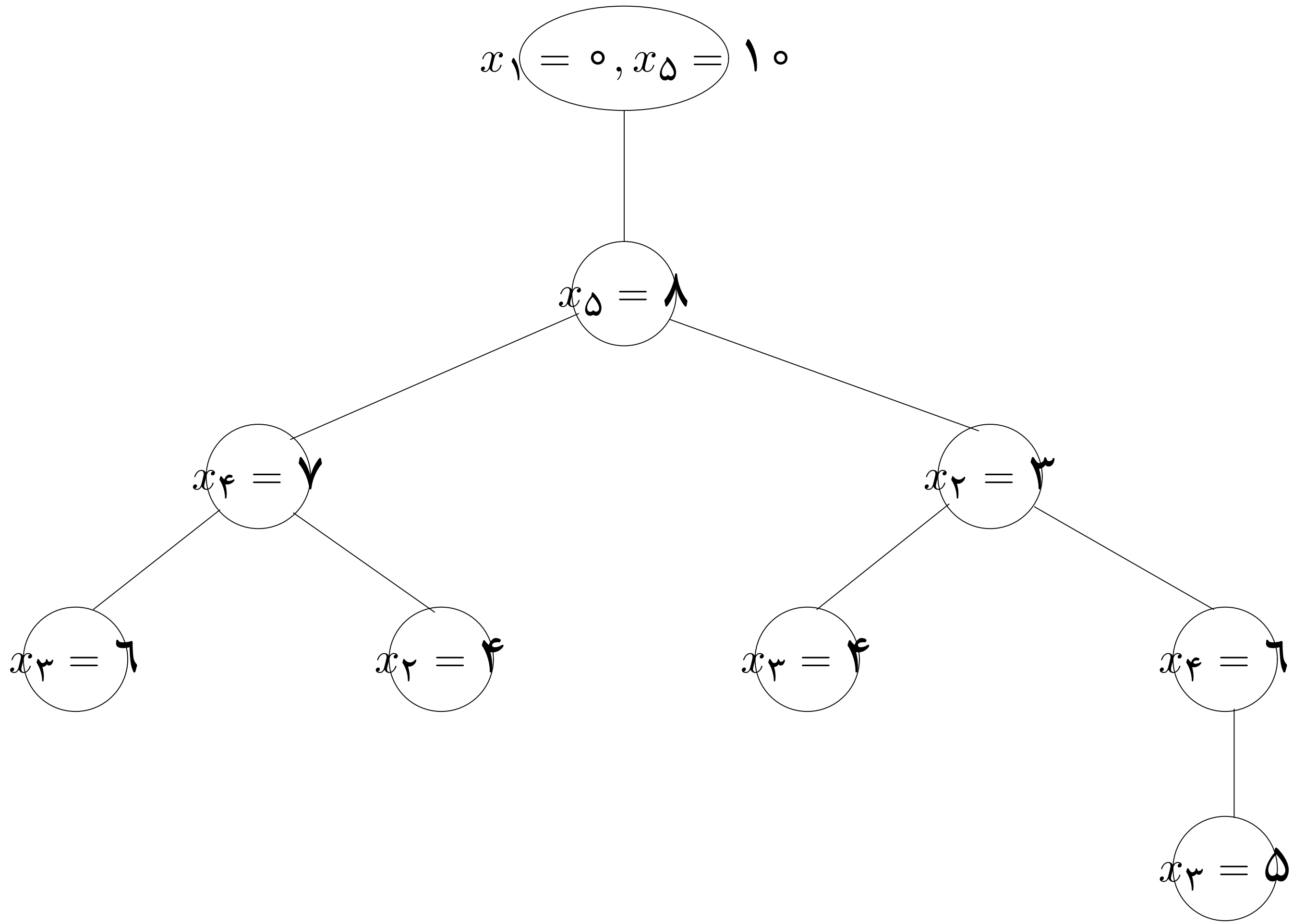
$$D = \{۱, ۲, ۳, ۵, ۵\}$$

پس از انتخاب $x_{۱}$، $x_{۶}$، $x_{۵}$، $x_{۲}$ و $x_{۴}$.



$$D = \{\}$$

پس از انتخاب x_1، x_6، x_5، x_2، x_4 و x_3.



درخت حالت

توجه

با توجه به این که اعداد D از بزرگ به کوچک مورد بررسی قرار می‌گیرند، برای تولید بزرگ‌ترین عدد $y \in D$، باید نقطه‌ی x_i را انتخاب کنیم که $y = x_i - x_1$ یا $y = x_n - x_i$ و لازم نیست با نقاط دیگر بررسی شود.

اگر با نقطه‌ی دیگری فاصله‌ی y تولید شود، فاصله x_i تا x_1 یا x_n بیش‌تر از y می‌شود که این ممکن نیست.

Turnpike(D, n)

$x_1 \leftarrow 0$
$x_n \leftarrow$ Deletemax(D)
$x_{n-1} \leftarrow$ Deletemax(D)
if $x_n - x_{n-1} \in D$
  then Delete$(x_n - x_{n-1}, D)$
    return Place$(D, 2, n-2, n)$
return **false**

PLACE$(D, left, right, n)$

if D is empty
 then **return** **true**
$dmax \leftarrow$ FINDMAX(D)
 ▷ CHECK IF $x_{right} = dmax$ IS FEASIBLE
if $(\forall\, 1 \leq j < left$ **and** $right < j \leq n, |x_j - dmax| \in D)$
 then $x_{right} \leftarrow dmax$
   for $1 \leq j < left$ **and** $right < j \leq n$
    do DELETE$(|x_j - dmax|, D)$
   $found \leftarrow$ PLACE$(D, left, right - 1, n)$
   if **not** $found$ {backtrack}
    then for $1 \leq j < left$ **and** $right < j \leq n$
      do INSERT$(|x_j - dmax|, D)$
if **not** $found$
 then try similarly to place $x_{left} \leftarrow x_n - dmax$
return FOUND

این الگوریتم از Skiena, Smith, and Lemke, 1990 **است.**

تحلیل

D را به‌صورت درخت بازه (درخت قرمز–سیاه) پیاده‌سازی می‌کنیم،

بررسی این‌که یک بازه در D هست یا خیر در $O(\lg n)$ انجام می‌شود

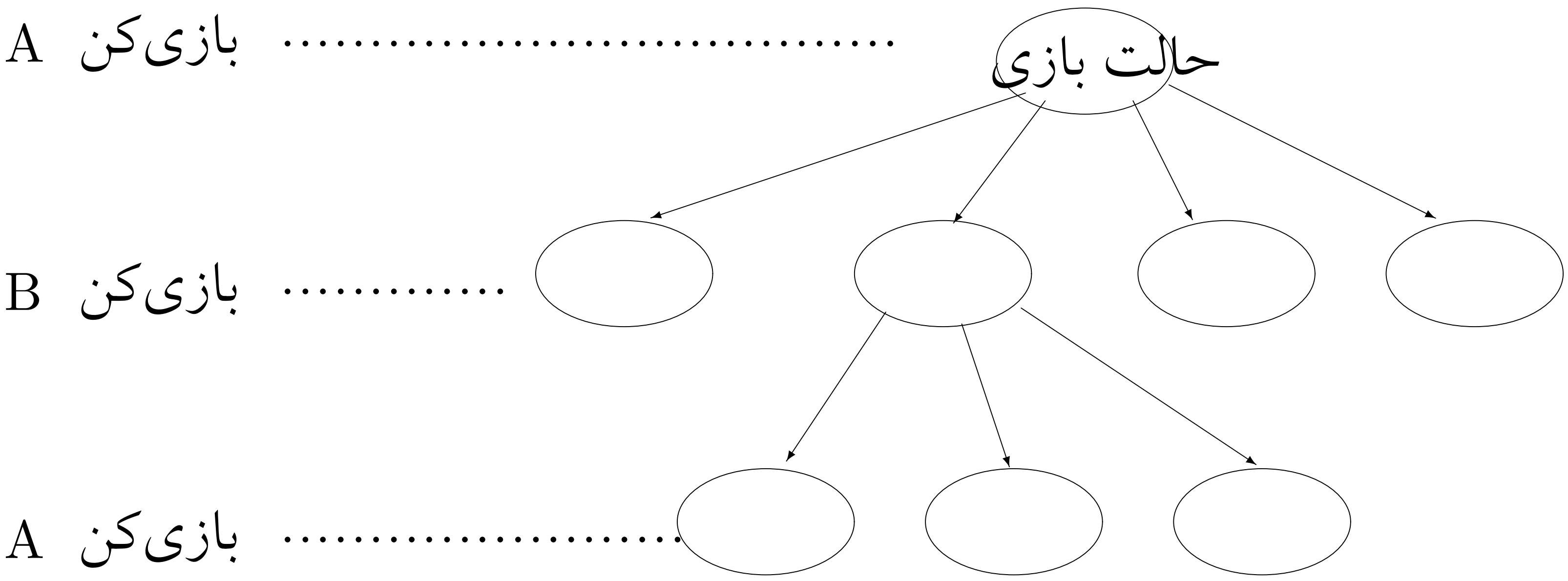
اگر پس‌گرد اتفاق نیقتد می‌توان این الگوریتم را در $O(n^۲ \lg n)$ انجام داد.

حداکثر تعداد پس‌گردها ۲^n است، پس در بدترین حالت الگوریتم $O(۲^n n \lg n)$ زمان می‌گیرد. و مثالی برای این حالت ادعا می‌شود.

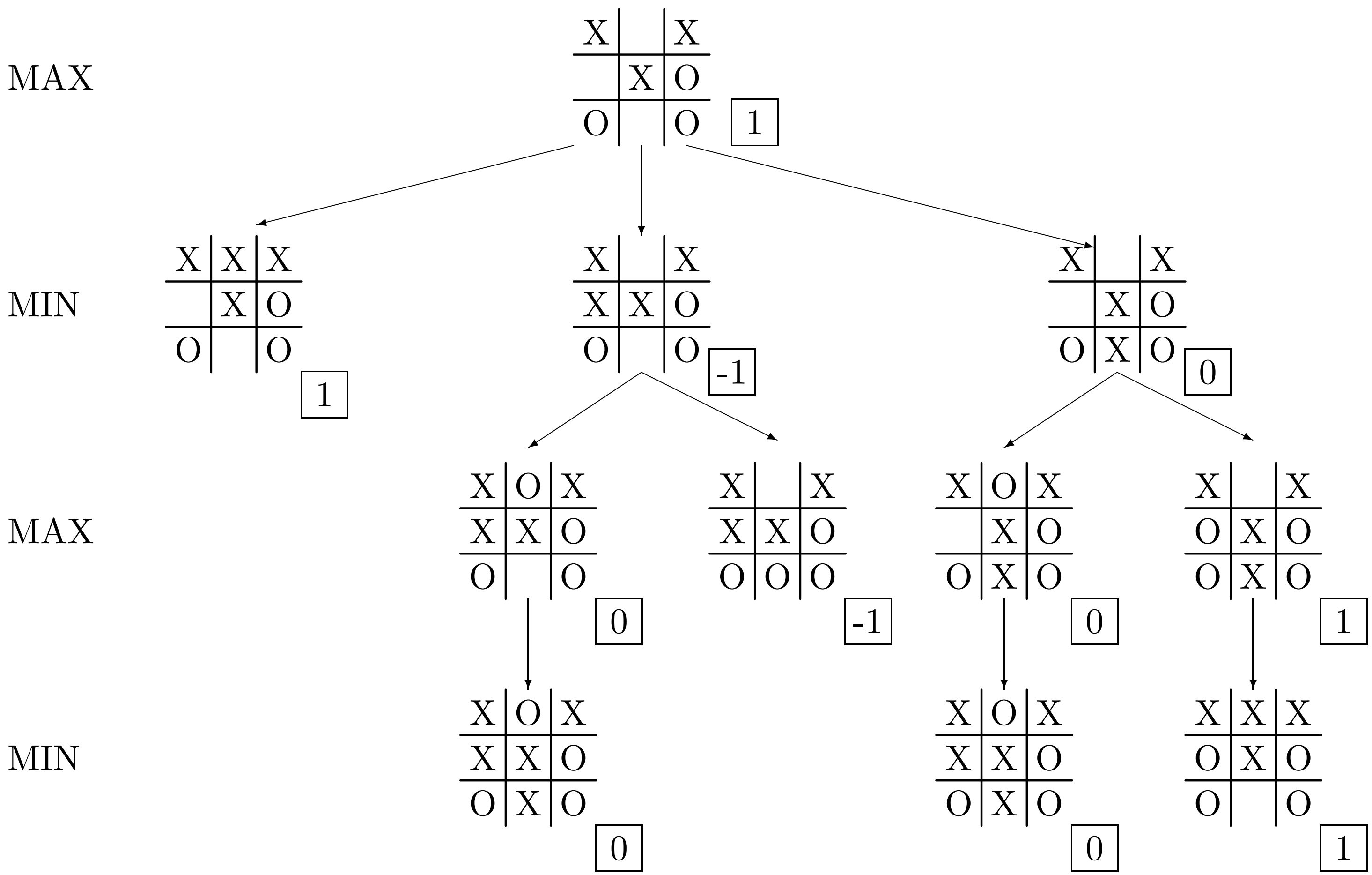
اگر اعداد صحیح و توزیع یک‌نواخت داشته باشند، با احتمال زیاد حداکثر یک پس‌گرد داریم.

حالت دوبعدی مسئله؟

درخت بازی



درخت بازی در بازی X-O


MAX
MIN
MAX
MIN

پیاده‌سازی درخت بازی

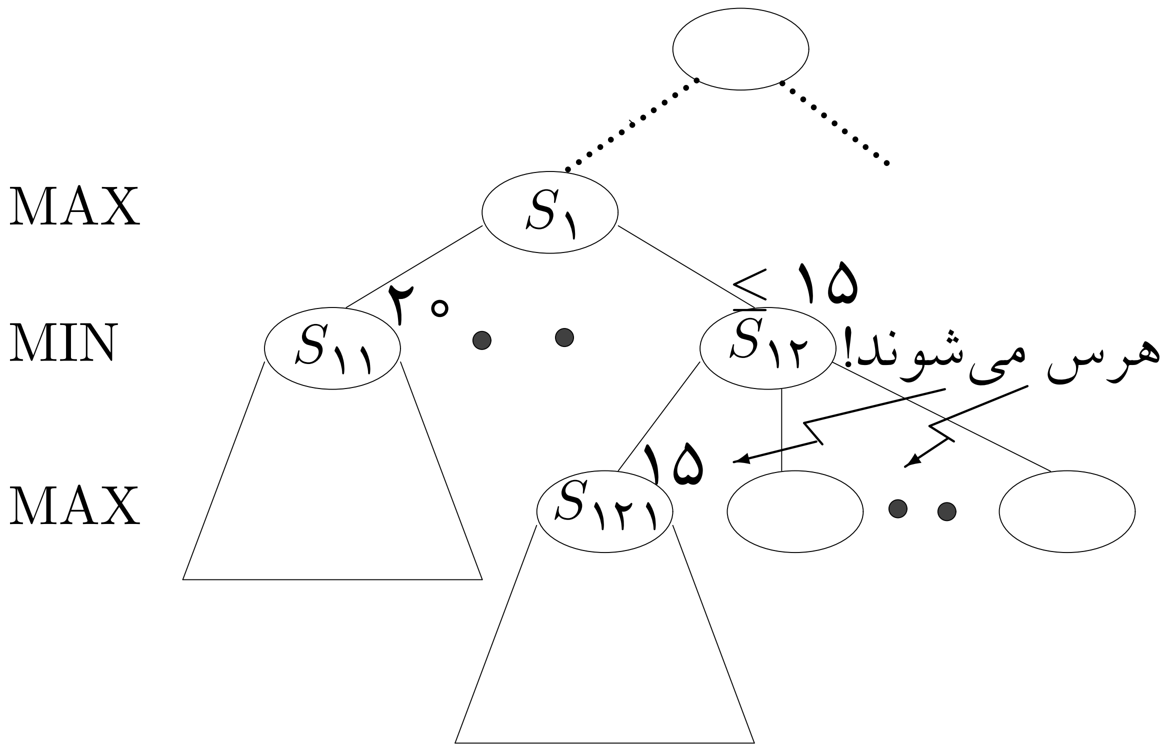
```
SEARCH(B, mode)
      ▷ B is is a board and mode is either max or min
      ▷ It returns a real value for maximum score
  if  B is a leaf
     then  return PAYOFF(B)
     else   if mode = Max
               then value ← −1
               else  value ← +1
            for  each child C of board B
                do if mode = max
                      then value ← max(value, SEARCH(C, min))
                      else  value ← min(value, SEARCH(C, max))
             return  value
```

محدود کردن فضای جستجو

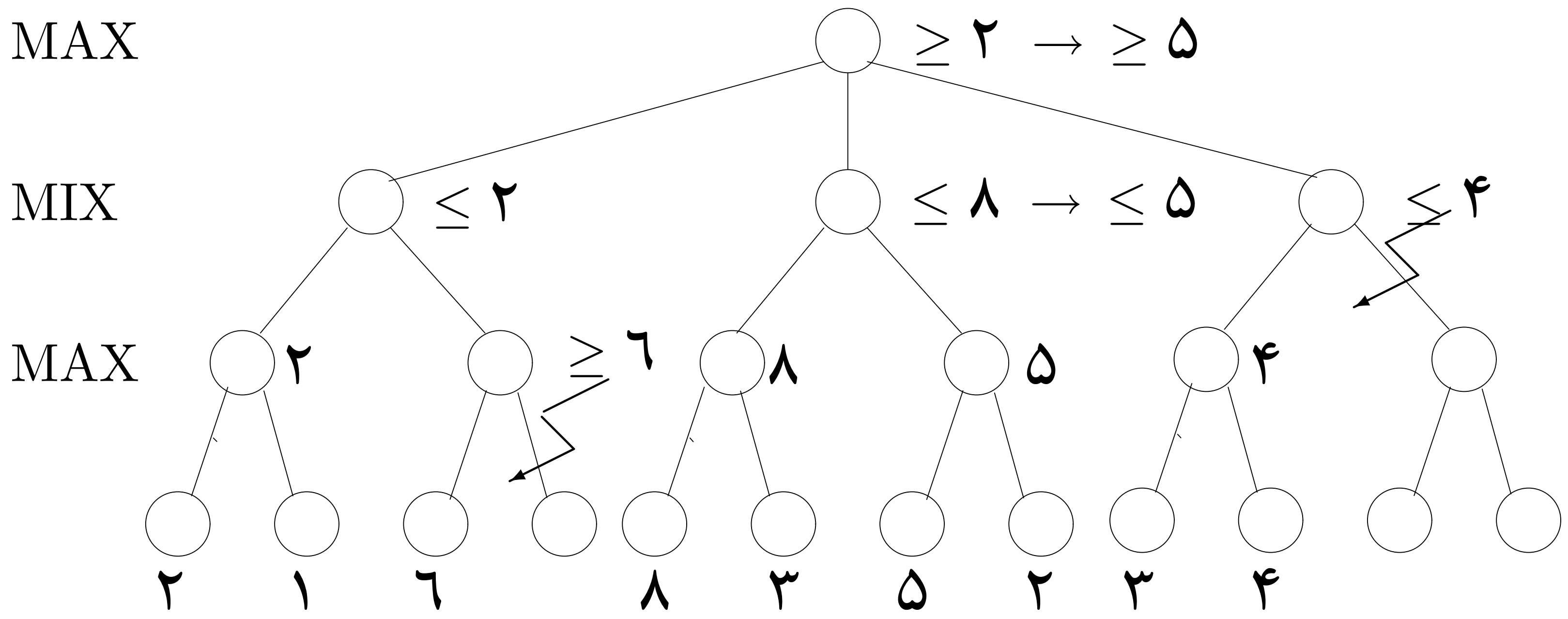
۱) هرس کردن (pruning)

۲) انشعاب و حد (Branch & Bound)

هرس کردن



با بازکردن $S_{۱۲۱}$ بی‌تاثیر بودن $S_{۲۱}$ مشخص می‌گردد و بقیه‌ی فرزندان آن هرس می‌شوند.



نمونه‌ای از یک α–β pruning

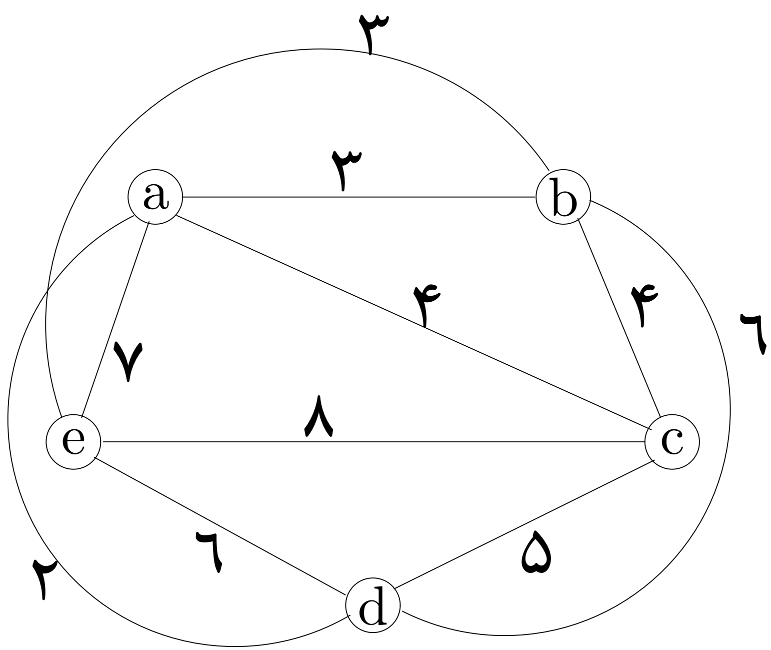
انشعاب و حد (Branch & Bound)

فروشنده‌ی دوره‌گرد (Travelling Salesman Problem)

فرض کنید شهرها و راه‌های و طول آن‌ها بین هر دو شهر داده شده است.

هدف پیدا کردن دوری با حداقل وزن ممکن است، که از کلیه‌ی شهرها فقط یک‌بار عبور کند.

یک مسئله‌ی ان‌پی-تمام!

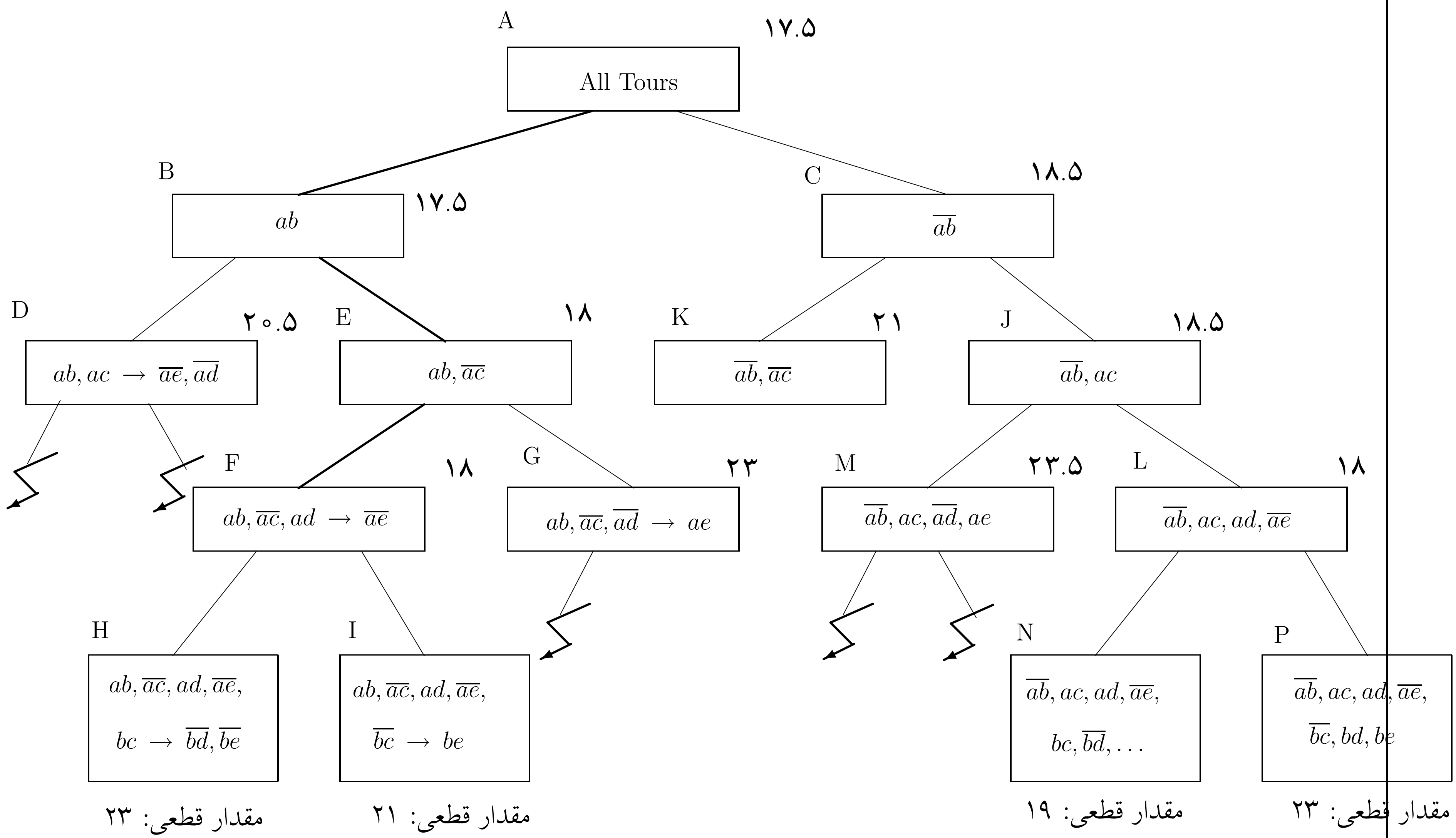


مسیر شهرها در مسئله‌ی فروشنده‌ی دوره‌گرد.

تعیین حد: در هر حالت برای هر رأس دو یال با کمترین وزن، از «یال‌های موجود» انتخاب می‌کنیم. بدیهی‌است که هیچ مسیر همیلتونی پیدا نمی‌شود که وزن آن کم‌تر از نصف مجموع وزن‌های این یال‌ها باشد. این حد همان اطلاعات محلی هر حالت محسوب می‌شود. مثلاً در حالتی که هیچ یالی حذف نشده است، حد به طریق زیر محاسبه می‌شود:

$$\begin{array}{ccc} a & : & ۲ + ۳ \\ b & : & ۳ + ۳ \\ c & : & ۴ + ۴ \\ d & : & ۲ + ۵ \\ e & : & ۳ + ٦ \\ \hline & & ۳۵ \div ۲ = ۱۷.۵ \end{array}$$

لذا هیچ مسیری با وزن کمتر از ۱۷/۵ و در واقع کم‌تر از ۱۸ نداریم (چون وزن‌ها عدد صحیح هستند.)


A
۱۷.۵
All Tours
B
ab
۱۷.۵
C
$\overline{ab}$
۱۸.۵
D
$ab, ac \rightarrow \overline{ae}, \overline{ad}$
۲۰.۵
E
$ab, \overline{ac}$
۱۸
K
$\overline{ab}, \overline{ac}$
۲۱
J
$\overline{ab}, ac$
۱۸.۵
F
$ab, \overline{ac}, ad \rightarrow \overline{ae}$
۱۸
G
$ab, \overline{ac}, \overline{ad} \rightarrow ae$
۲۳
M
$\overline{ab}, ac, \overline{ad}, ae$
۲۳.۵
L
$\overline{ab}, ac, ad, \overline{ae}$
۱۸
H
$ab, \overline{ac}, ad, \overline{ae},$
$bc \rightarrow \overline{bd}, \overline{be}$
I
$ab, \overline{ac}, ad, \overline{ae},$
$\overline{bc} \rightarrow be$
N
$\overline{ab}, ac, ad, \overline{ae},$
$bc, \overline{bd}, \ldots$
P
$\overline{ab}, ac, ad, \overline{ae},$
$\overline{bc}, bd, be$
مقدار قطعی: ۲۳
مقدار قطعی: ۲۱
مقدار قطعی: ۱۹
مقدار قطعی: ۲۳