DATE / 400109843 SUBJECT : پویا آهی

سوال ۱)

آ) در مسائل CSP برای انتخاب کردن اینکه کدام node را باید مورد بررسی قرار دهیم می‌توانیم policy های مختلف داشته باشیم که در ادامه به صورت خلاصه دو مورد از این policy ها را توضیح می‌دهم

MRV (minimum remaining values):

برای اینکه انتخاب کنیم در مرحله‌ی بعدی کدام variable را مقداردهی کنیم. به سراغ variable ای می‌رویم که کمترین تعداد حالات ممکن بین بقیه را دارد و دلیل انجام این کار، به این دلیل است که احتمال اینکه این variable زودتر، انتخابات ممکن آن صفر شود بیشتر است بنابراین زودتر آن را مقدار دهی می‌کنیم تا احتمال backtrack کمتری داشته باشیم.

LCV (least Constraining Value):

در این policy ابتدا variable هایی را مقدار دهی می‌کنیم که در همسایه‌های خود کمترین تغییرات را داشته باشد. در این policy، به دلیل اینکه از forward checking یا arc-consistency های بیشتری باید استفاده کنیم. نیاز به محاسبات بیشتری هست اما همچنان با سرعت بیشتری به جواب می‌رسیم.

NOTEBOOK iman

CamScanner

DATE / / SUBJECT :

ب)

در این گراف اگر رأس وسط را انتخاب کنیم و با ~~استفاده~~ از ایده‌ی cutset Conditioning استفاده کنیم به جواب و راه حل بهینه‌ای می‌رسیم.

به صورت خلاصه با انتخاب کردن رأس وسط تمامی value های ممکن را به آن رأس می‌دهیم و به ازای هر value مسئله‌ی جدید را حل می‌کنیم. این کار به این صورت انجام می‌شود که با ~~حذف~~ انتخاب کردن value برای رأس وسط می‌توانیم با اعمال کردن شروط آن رأس با رأس های همسایه، value های غیرممکن را از همسایه‌های این رأس حذف کنیم و به صورت کلی این رأس را از مسئله حذف کنیم. در این صورت مسئله به 4 زیر درخت تقسیم می‌شود که حل کردن مسائل tree-structured CSP به سادگی قابل انجام است (در زمان $nd^2$)

ج)

خیر، با توجه به اینکه بازی non-zero sum است. بازیکن ها به دنبال min یا max کردن یک value ~~مشترک~~ نیستند و هر بازیکن به دنبال max کردن value خود می‌باشد. به همین دلیل نمی‌توانیم از alpha-beta استفاده کنیم.

NOTEBOOK iman

CamScanner

د)

- ~~استفاده~~ در expectimax لزوماً به یک جواب optimal نمی‌رسیم، احتمال این است که در نهایت در state ای قرار بگیریم که احتمال بردن ۱۰ میلیون تومان یا بیش از آن کاهش پیدا کرده باشد.
- اگر از minimax استفاده کنیم حتی اگر حریف ما Adverserial باشد. ما تصمیمی می‌گیریم که لزوماً احتمال را افزایش دهد. با توجه به اینکه حریف حرکت random انجام می‌دهد. ممکن است تصمیمی که می‌گیریم احتمال بردن را به بالاترین حد ممکن نرساند اما هدف الگوریتم ~~افزایش این احتمال~~ ماکس کردن این احتمال است.
- در این قسمت با توجه به اینکه درخت به سمتی حرکت می‌کند که مقدار Z شود، در واقع هر دو درخت expectimax و minimax در حال تلاش برای افزایش این احتمال هستند و در هر دو درخت به یک جواب optimal می‌رسیم.

NOTEBOOK iman

CamScanner

DATE / / SUBJECT :

سوال 2)

آ) متغیرها در این مسئله (تکالیف) Hها هستند.

و دامنه هر یک از این Hها Time شروع آنها است.

$T = \{0, 1, 2, \dots, 14\}$.

متغیرها: $\{H_{00}, H_{01}, H_{02}, H_{03}, H_{10}, H_{11}, H_{12}, H_{20}, H_{21}, H_{22}, H_{23},$

$H_{30}, H_{31}, H_{32}, H_{33}\}$

ب) در این مسئله CSP، طبق قسمت قبلی متغیرها و دامنه‌ها را

تعریف کردیم و در این بخش تنها لازم است که قیود را قرار دهیم.

$H_{00}, H_{01}, H_{02}, H_{10} \rightarrow$ Computer 0

$H_{11}, H_{20}, H_{22} \rightarrow$ Computer 1

$H_{21}, H_{22}, H_{31}, H_{32} \rightarrow$ Computer 2

$H_{03}, H_{10}, H_{12}, H_{23}, H_{30}, H_{30} \rightarrow$ No Computer Needed

* هر کامپیوتر در یک زمان یک ~~تکلیف~~ کار می‌تواند انجام دهد.

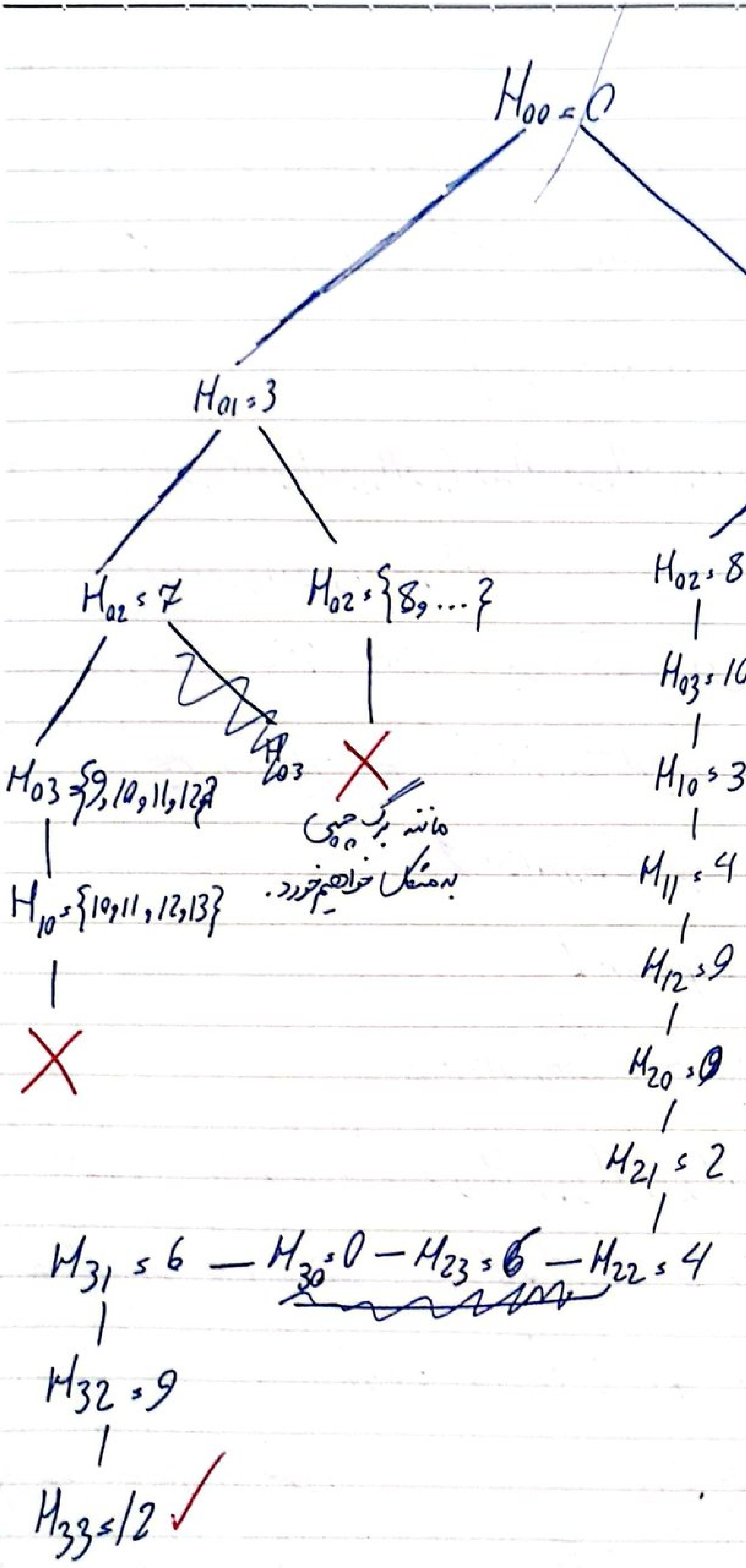
Computer $i$ : $T_{H_i} + H_i \leq H_j : \forall i, j$

* تایم بیشتر از 14 نشود: $\forall i : T_{H_i} + H_i \leq 14$

* برای هر کلاس تمارین به ترتیب باید انجام شوند. $\forall i, j : \cancel{T_{H_{ij}}} \leq T_{H_{ij}} \leq T_{H_{ij+1}}$

NOTEBOOK iman

CamScanner


$H_{00} = 0$
$H_{01} = 3$
$H_{01} = 4$
$H_{02} = 7$
$H_{02} = \{8, \ldots\}$
$H_{02} = 8$
$H_{03} = \{9, 10, 11, 12\}$
$H_{03}$
$H_{03} = 10$
$H_{10} = \{10, 11, 12, 13\}$
مانند برگ چپی به مشکل خواهیم خورد.
$H_{10} = 3$
$H_{11} = 4$
$H_{12} = 9$
$H_{20} = 0$
$H_{21} = 2$
$H_{31} = 6 - H_{30} = 0 - H_{23} = 6 - H_{22} = 4$
$H_{32} = 9$
$H_{33} = 12$

DATE / / SUBJECT :

در این مسئله ابتدا از بالا چپ شروع کردیم و پایین رفتیم.

د) در این سوال علاوه بر استفاده از LCV و MRV از Forward-Checking نیز استفاده می‌کنیم (بدین معنا که با قرار دادن هر Variable) حالت دامنهٔ متغیرهای همسایه را تغییر می‌دهیم.

Start
$H_{11} = 1$
$H_{10} = 0$
$H_{12} = 6$
$H_{20} = 6$
$H_{21} = 8$
$H_{22} = 10$
$H_{23} = 12$
$H_{31} = 2$
$H_{30} = 0$
$H_{32} = 5$
$H_{33} = 8$
$H_{01} = 3$
$H_{00} = 1$
$H_{02} = 8$
$H_{03} = 10$
~~$H_{04} =$~~

هـ) با توجه به درخت های جستجو در قسمت ج و د)

درخت جستجوی که در قسمت د) داشتیم، اندازه کمتری داشته، که دلیل آن

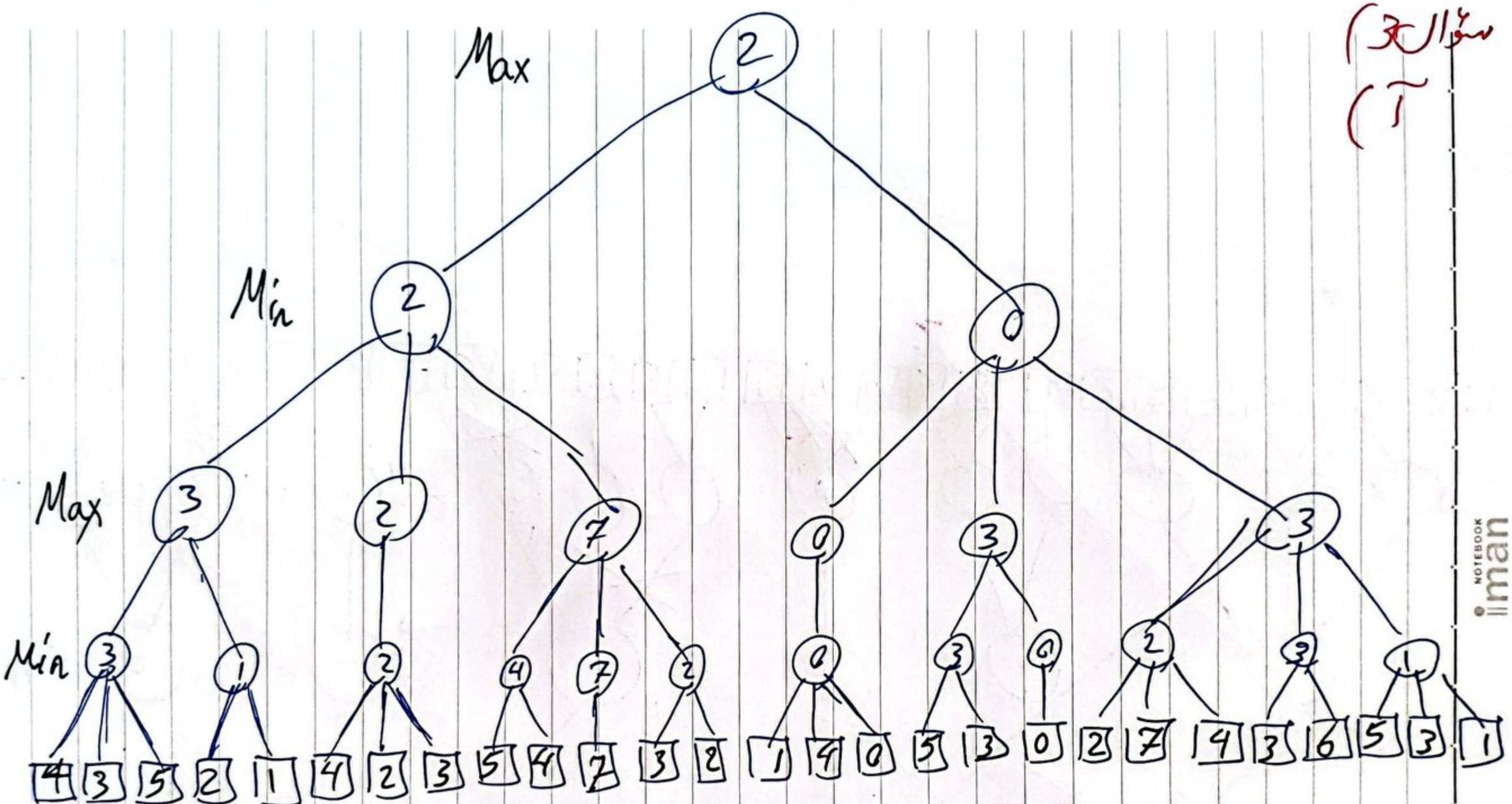
هم استفاده از heuristic های مانند LCV و MRV است.

در قسمت ج بدون استفاده از هیچ heuristic ای عمل کردیم که

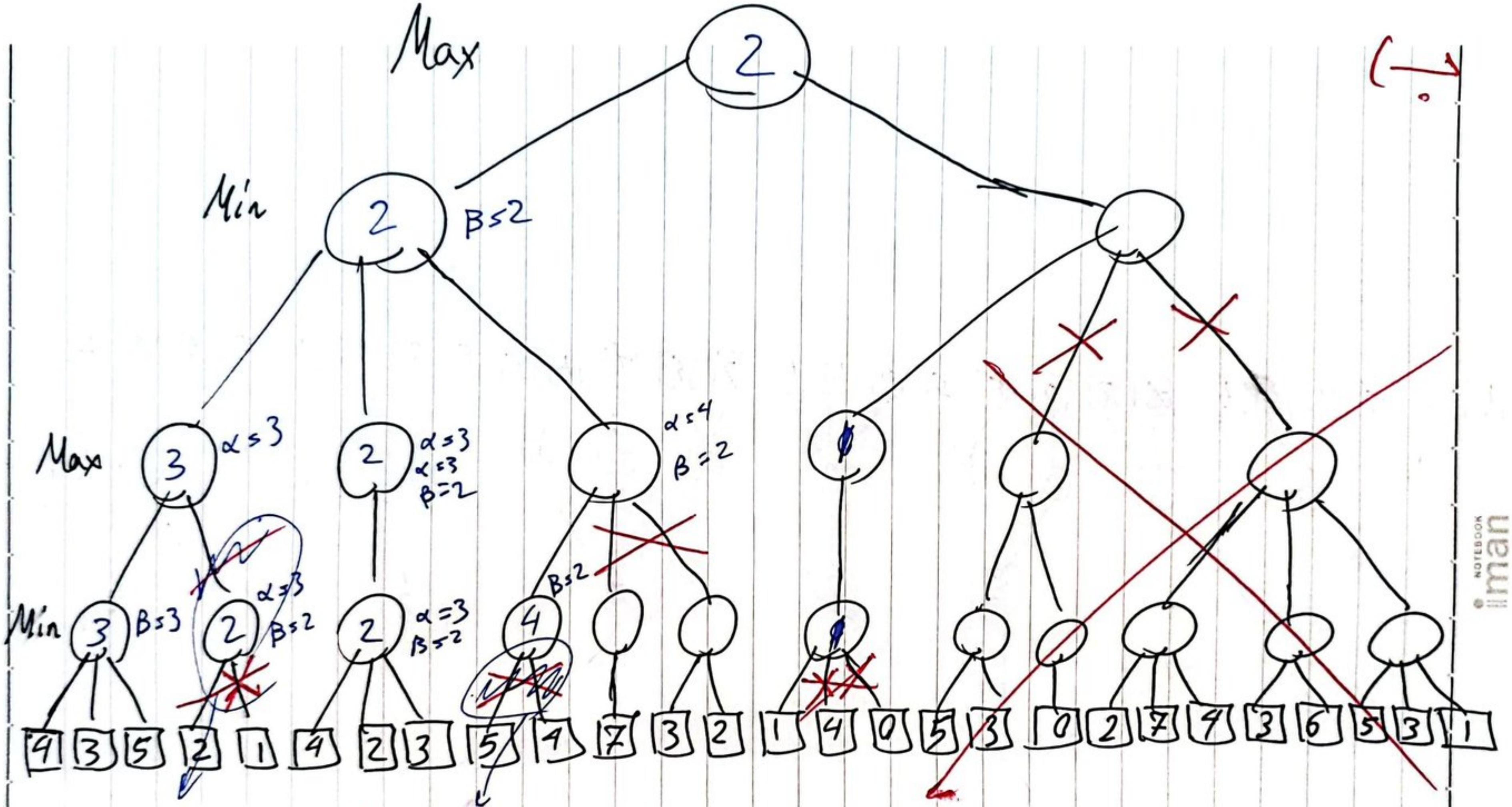
این اتفاق باعث شد تا اینکه نیاز به backtrack های بیشتری داشته باشیم

تا به جواب درست برسیم.

(سوال 3
(آ

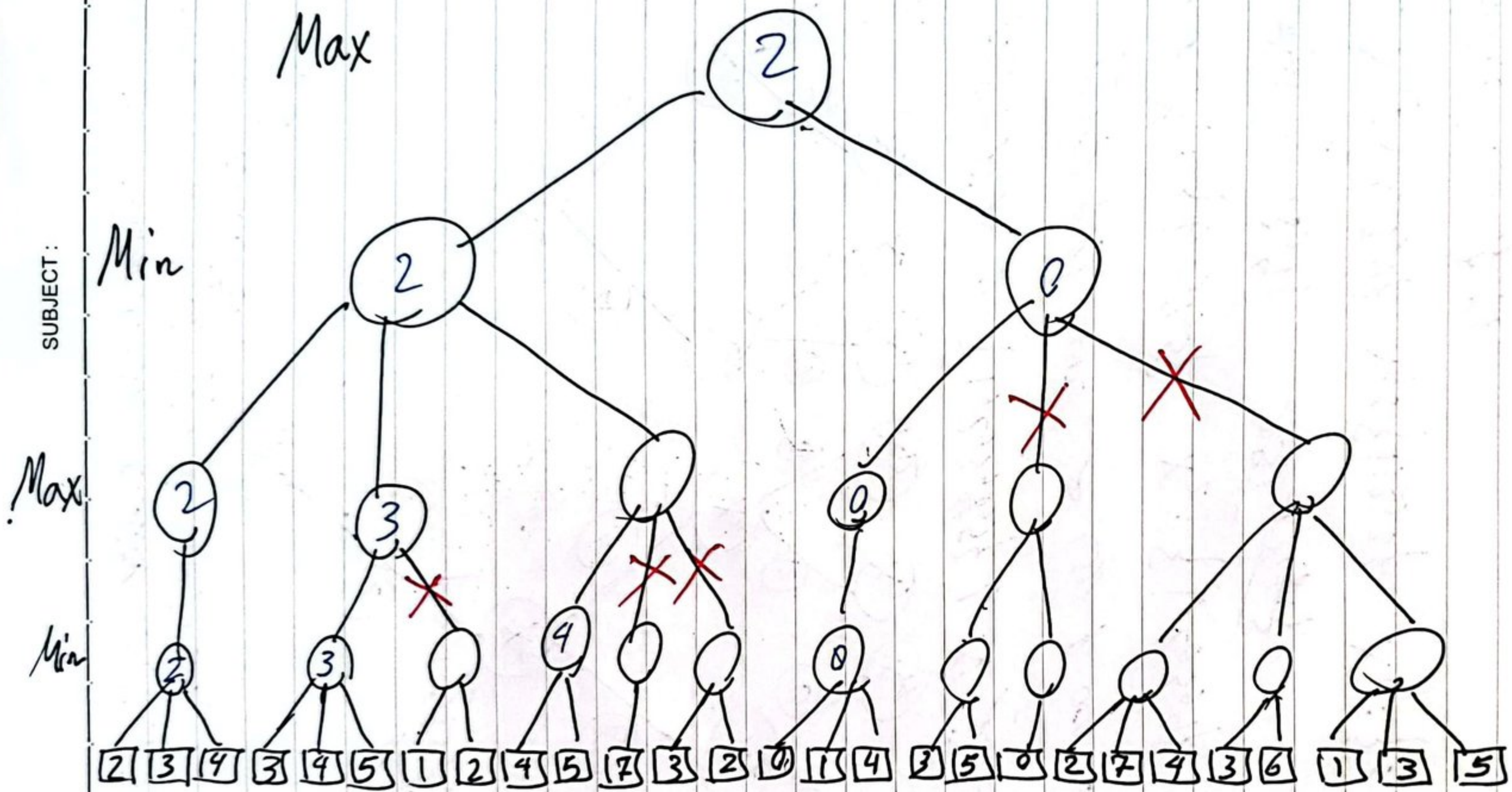



SUBJECT :
DATE / /
iman NOTEBOOK
CamScanner




prune نمی‌شود

prune نمی‌شود



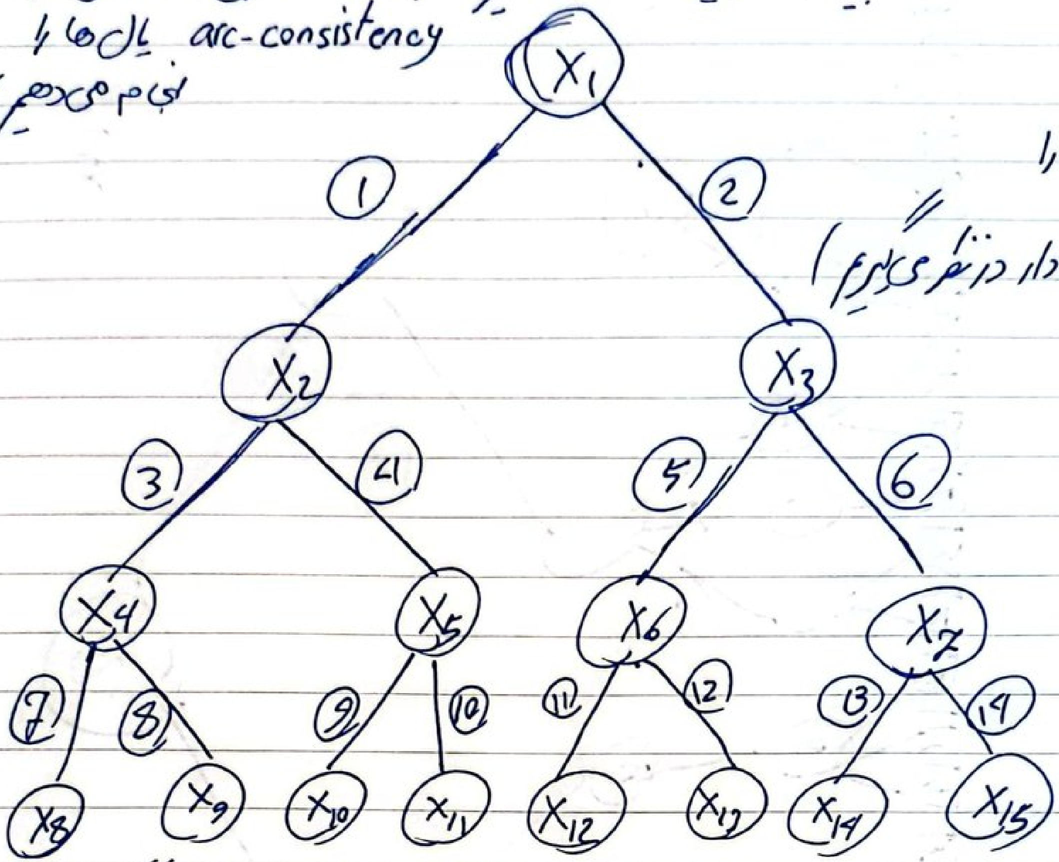
Max
Min
Max
Min

DATE / / SUBJECT :

سوال ۶) آ)

(لازم به ذکر است در arc-consistency، بعد از هر بار Consistent کردن یال ها
را باید در آخر لیست اضافه کنیم اما برای ساده سازی در همان مرحله
arc-consistency یال ها را
انجام می دهیم)

(هر یک arc را
دو یال جهت دار در نظر می گیریم)




در ابتدا دامنه هر متغیر $X_i$ را $\{1,2,3,4,5\}$ در نظر می گیریم

و در هر مرحله arc consistency را روی هر arc انجام می دهیم.

① → $X_1 : \{2,3,4,5\}$ $X_2 : \{1,2,3,4\}$

② → $X_3 : \{1,2,3,4\}$ ~~③ → $X_4 : \{\}$~~

③ → $X_2 : \{2,3,4\}$ $X_4 : \{1,2,3\}$ $X_1 : \{3,4,5\}$

④ → $X_5 : \{1,2,3\}$

⑥ → $X_7 : \{1,2,3\}$

⑤ → $X_6 : \{1,2,3\}$ $X_3 : \{2,3,4\}$

iman NOTEBOOK

CamScanner

DATE / /        SUBJECT :

(7) $\rightarrow X_8 : \{1,2\} \quad X_4 : \{2,3\} \quad X_2 : \{3,4\} \quad X_1 : \{4,5\}$

(8) $\rightarrow X_9 : \{1,2\}$ (9) $\rightarrow X_{10} : \{1,2\} \quad X_5 : \{2,3\}$

(10) $\rightarrow X_{11} : \{1,2\}$

(11) $\rightarrow X_{12} : \{1,2\} \quad X_6 : \{2,3\} \quad X_3 : \{3,4\}$

(12) $\rightarrow X_{13} : \{1,2\}$ (13) $\rightarrow X_{14} : \{1,2\} \quad X_7 : \{2,3\}$

(14) $\rightarrow X_{15} : \{1,2\}$

در نهایت $X_i$ ها دامنه آنها مانند زیر می‌شود:

$X_1 : \{4,5\} \qquad X_2 : X_3 : \{3,4\}$

$X_4 : X_5 : X_6 : X_7 : \{2,3\}$

$X_8 : X_9 : X_{10} : X_{11} : X_{12} : X_{13} : X_{14} : X_{15} : \{1,2\}$

ب)

بله این شبکه Consistent هست.

یک جواب ممکن:

$X_1 = 5 \qquad X_2 = X_3 = 4$

$X_4 = X_5 = X_6 = X_7 = 3$

$X_8 = X_9 = X_{10} = X_{11} = X_{12} = X_{13} = X_{14} = X_{15} = 2$

CamScanner

DATE / / SUBJECT :

ج) برای حل این مسئله با توجه به اینکه این درخت Consistent است.

برای هر یک مقدار $X_i$ برای فرزندان آن لزوماً یک جواب داریم.

با استفاده از arc-consistency دامنهٔ درست را مشخص می‌کنیم و شروع به مقداردهی

می‌کنیم. در این صورت تنها با چک کردن قید بین پدر و فرزند

می‌توانیم به جواب درست برای مسئله ~~برسیم~~ بدهیم.

د)

در مسائل CSP به صورت درختی اگر در پیچیدگی مسئله به صورت $O(Nd^2)$

است که در آن $N$ تعداد متغیر و $d$ تعداد اعضای درون دامنهٔ آن است.

با چک کردن درخت از بالا یا پایین تمامی برگ‌ها را یک بار چک می‌کنیم

که در $O(N)$ انجام می‌شود و در بدترین حالت در $O(d^2)$ باید جواب‌ها تک تک

pair ها را چک کنیم.

در نهایت پیچیدگی این الگوریتم $O(Nd^2)$ است.

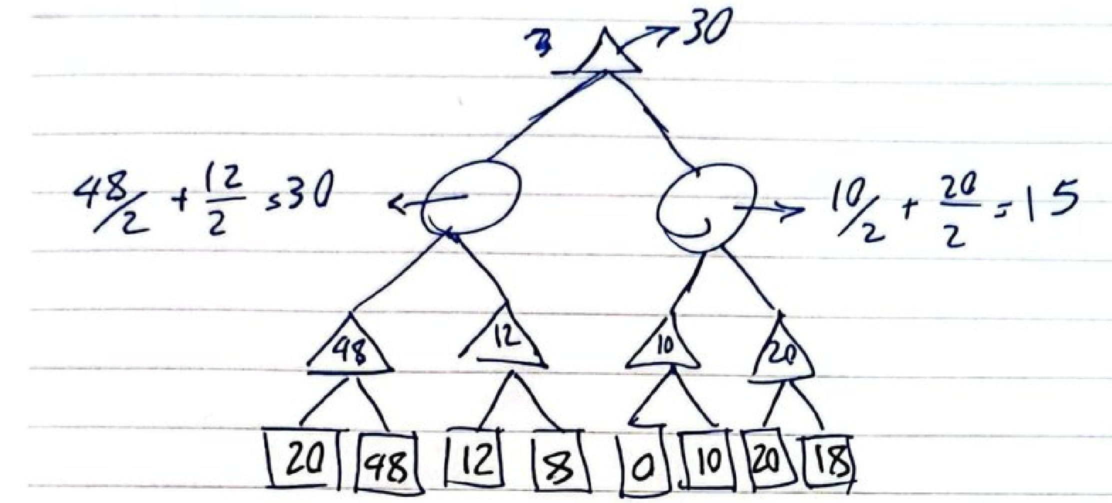


CamScanner

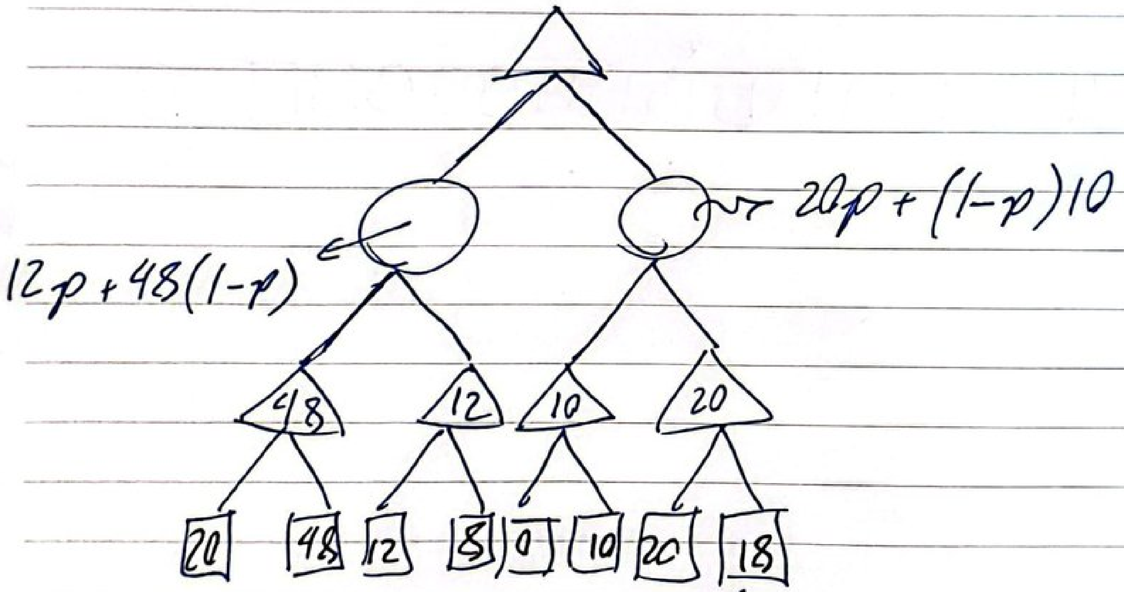
DATE / /  SUBJECT :

سوال ۶)

آ)




ب)




برای اینکه حرکت optimal تغییر کند ریشه باید شاخه راستی را انتخاب کند.

$$20p+(1-p)10 > 12p+48(1-p) \Rightarrow 10p+10 > 48-36p \Rightarrow$$

$$\Rightarrow 46p > 38 \Rightarrow p > \frac{38}{46}$$

CamScanner