هوش مصنوعی

دانشکده مهندسی کامپیوتر

دکتر رهبان
بهار ۱۴۰۳

مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶

تمرین دوم، تئوری

سوال ۱

۱. (۱۵ نمره، درجه سختی ۷) درستی یا نادرستی عبارت‌های زیر را با ذکر دلیل مشخص کنید:

الف) حداکثر تعداد دفعاتی که الگوریتم backtracking ممکن است مجبور به backtrack شود، اگر از arc consistency و هیوریستیک‌های MRV و LCV استفاده کند $O(dn^2)$ است.(n تعداد متغیرها و d تعداد مقادیر مجاز برای هر متغیر است.)

ب) اگر گراف محدودیت یک مسئله CSP با محدودیت های دودویی به صورت درخت با n رأس باشد، پیچیدگی محاسباتی حل کننده کارا برحسب n ، $O(n^2)$ است.

ج) الگوریتم هرس آلفابتا علاوه بر آنکه زمان را کاهش میدهد در جواب به دست آمده برای ریشه درخت با minimax نیز تأثیرگذار می‌باشد.

برای دو مورد بعدی تابع اکیدا صعودی F و یک بازی zero-sum با دو بازیکن را درنظر بگیرید:

د) اعمال تابع F روی برگ‌های یک درخت minimax برای این بازی پاسخ بهینه آن را تغییر نخواهد داد.

ه) اعمال تابع F روی برگ‌های یک درخت minimax برای این بازی برگ‌هایی که توسط alpha-beta pruning هرس می شوند را تغییر نمی دهد.

(آ) نادرست. استفاده از این توابع فقط باعث می شود که این الگوریتم به طور بهینه تری جست و جو می کند ولی همچنان ممکن است worst case اتفاق بیافتد، پس نمی توان گفت اردر زمانی تغییری می کند و همان d^n است.

(ب) نادرست. اردر زمانی حل کردن یک مسئله ی CSP از $O(nd^2)$ است. اگر گراف یک درخت دودویی باشد، $d = ۲$ و پیچیدگی حل از اردر n می شود

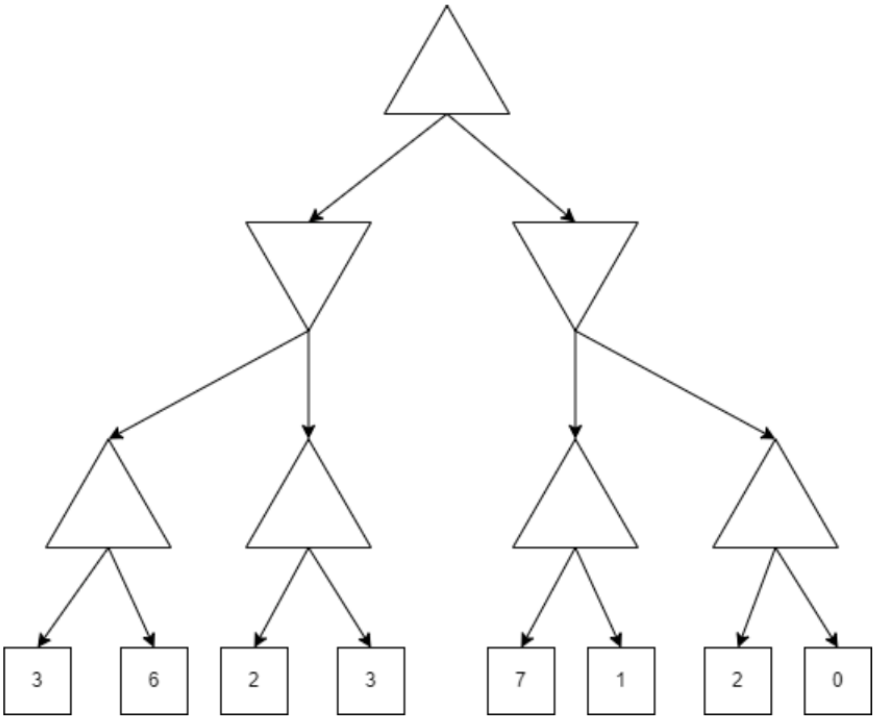
(ج) نادرست است. این الگوریتم فقط راس هایی را از درخت حذف می کند که مطمئن است قرار نیست بر روی نتیجه ی نهایی تاثیر بگذارند.

(د) درست است. می دانیم اگر داشته باشیم $x_1 < x_2$ با اعمال تابع اکیدا صعودی F جهت نامساوی تغییری نمی کند. $F(x_1) < F(x_2)$

(ه) درست است. به همان دلیل مورد د

سوال ۲

۲. (۱۵ نمره، درجه سختی ۵) درخت بازی زیر را در نظر بگیرید و به سوالات مربوطه پاسخ دهید:



شکل ۱

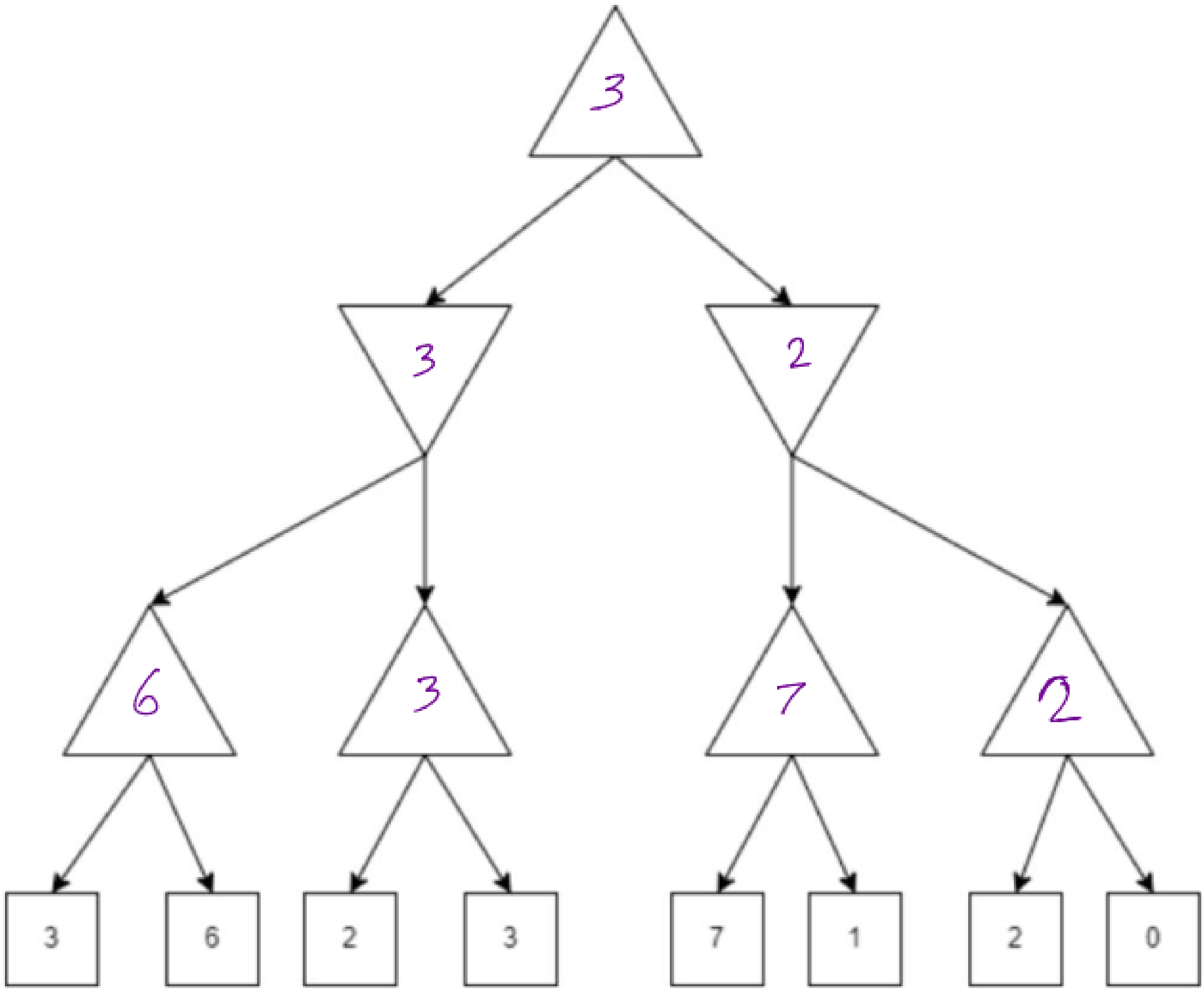
الف) درخت minimax بازی مورد نظر را کامل نمایید.

فرض کنید بازیکن اول یک قابلیت ویژه دریافت کند. این قابلیت ویژه این است که بازیکن اول می تواند با پرداخت کردن هزینه c حرکت انتخابی توسط بازیکن مقابل را تحت کنترل خود دربیاورد.

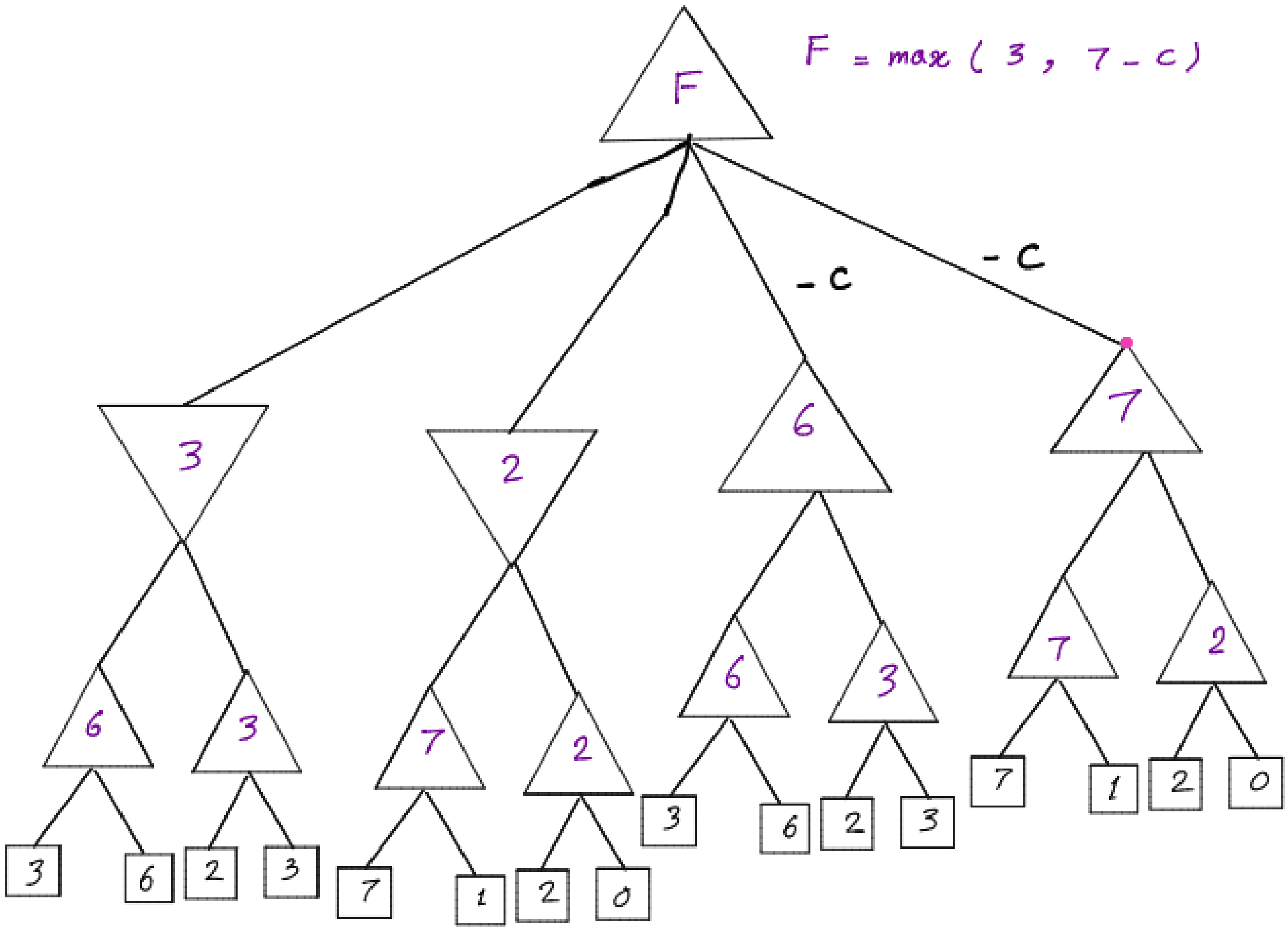
ب) با در نظر گرفتن فرض $c = ۲$ آیا برای بازیکن اول به صرفه خواهد بود که از این قابلیت ویژه استفاده کند؟ درخت بازی را مجدد رسم کرده و کامل نمایید. اگر پاسخ سوال قبل مثبت بود، نقطه ای در درخت که برای بازیکن اول بهینه است از قابلیت ویژه خود استفاده کند را نیز مشخص نمایید.

ج) بخش قبل را مجددا اما این بار با فرض $c = ۵$ روی هزینه قابلیت ویژه برای بازیکن اول پاسخ دهید.

(آ) .



(ب) .



همانطور که می بینیم با $c = ۲$ خروجی در حالتی که از قابلیت استفاده کنیم برای ما بهتر خواهد بود. نقطه ی استفاده نیز با نقطه ی صورتی مشخص شده.

(ج) در صورتی که این هزینه به ۵ افزایش یابد دیگر به صرفه نخواهد بود.

سوال ۳

۳. (۲۰ نمره، درجه سختی ۶) فرض کنید یک جدول به شکل زیر داریم که در هر خانه آن یک عدد تا یک رقم اعشار، در پایین هر ستون و روبه‌روی هر ردیف یک عدد صحیح نوشته شده است. حال می‌خواهیم اعداد درون جدول را به گونه ای به سمت بالا یا پایین گرد کنیم که مجموع اعداد هر ردیف با عدد روبه‌روی آن و مجموع اعداد هر ستون با عدد پایین آن برابر شود.

۴.۶	۵.۷	۳.۷	۱۴
۱.۵	۱.۶	۳.۷	۷
۷	۶	۸	

الف) این مسئله را به یک مسئله CSP تبدیل کنید.

ب) گراف محدودیت‌های آن را رسم کنید.

ج) با استفاده از روش forward checking و هیوریستیک‌های MRV و Degree با Backtrack مسئله را حل کنید.

(آ) .

X_1	X_2	X_3	14
X_4	X_5	X_6	7
7	6	8	

$D_1 = \{4, 5\}$ $\quad D_2 = \{5, 6\}$

$D_3 = \{3, 4\}$ $\quad D_4 = \{1, 2\}$

$D_5 = \{1, 2\}$ $\quad D_6 = \{3, 4\}$

$$f_1(X_1, X_4) = (X_1 + X_4 = 7)$$

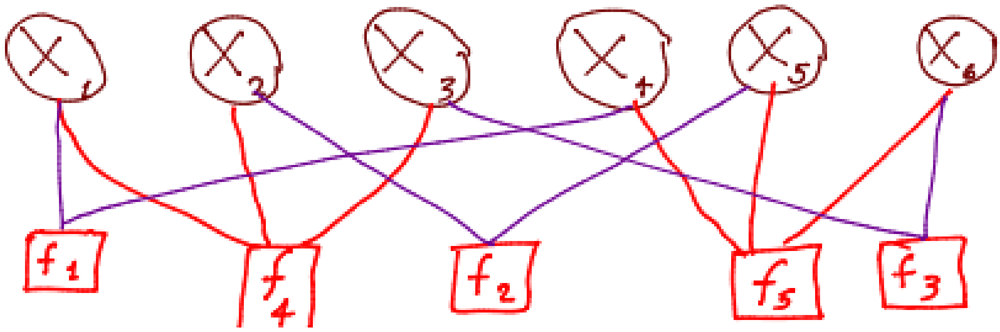
$$f_2(X_2, X_5) = (X_2 + X_5 = 6)$$

$$f_3(X_3, X_6) = (X_3 + X_6 = 8)$$

$$f_4(X_1, X_2, X_3) = (X_1 + X_2 + X_3 = 14)$$

$$f_5(X_4, X_5, X_6) = (X_4 + X_5 + X_6 = 7)$$

(ب) .



(ج).

Bactracking start:

MRV → $\{X_1, \dots, X_6\}$

degree → $\{X_1, \dots, X_6\}$ → X_1

$X_1 = 4$

forward checking → $D_1 = \{4\}$, $D_4 = \emptyset$

backtrack → $X_1 = \emptyset$

$X_1 = 5$

forward checking → $D_1 = \{5\}$, $D_4 = \{2\}$

MRV → $\{X_4\}$

$X_4 = 2$

forward checking → X

MRV → $\{X_2, X_3, X_5, X_6\}$

degree → 〃 → X_2

$X_2 = 5$

forward checking → $D_2 = \{5\}$, $D_3 = \{4\}$

$D_5 = \{1\}$

MRV = $\{X_3, X_5\}$

degree = 〃 → X_3

$X_3 = 4$

forwad checking → $D_6 = \{4\}$

MRV = $\{X_5, X_6\}$

degree = 〃 → X_5

$X_5 = 1$

forward checking → X

MRV = $\{X_6\}$ → X_6

$X_6 = 4$

$X_1 = 5$, $X_2 = 5$, $X_3 = 4$, $X_4 = 2$, $X_5 = 1$, $X_6 = 4$

سوال ۴

۴. (۲۰ نمره، درجه سختی ۷)

الف) با فرض آنکه $f, g : \mathbb{R} \to \mathbb{R}$ محدب‌اند و $t \geq ۰$ نشان دهید که توابع زیر محدب هستند یا خیر.(در صورت محدب بودن اثبات کنید درغیراینصورت مثال نقض ارائه دهید.)

- $h(x) = f(x) + tg(x)$
- $k(x) = max\{f(x), g(x)\}$
- $r(x) = min\{f(x), g(x)\}$
- $s(x) = f(x)g(x)$

ب) تعیین کنید که مجموعه C که برای زوج مرتب‌های شامل یک بردار x و عدد حقیقی t به شکل زیر تعریف می‌شود یک مجموعه محدب می‌باشد یا خیر.

$$C = \{(x, t) \mid \|x\| \leq t\}$$

(آ)
- محدب است.

$$h''(x) = f''(x) + tg''(x) \xrightarrow{\text{f and g are convex}} h''(x) \geq ۰, \text{h is convex}$$

- محدب است.

$$k(x) = \begin{cases} f(x) & f(x) > g(x) \\ g(x) & \text{o.w} \end{cases}$$

واضح است که در طول بازه ها، مشتق دوم همواره مثبت است. تنها کافیست در شکستگی ها بررسی کنیم که تابعی که قرار است خروجی k به آن تغییر کند دارای شیب بیشتری باشد که خب در حالت max این گونه هست، درغیر این صورت خروجی تابع اصلا عوض نمیشد

- محدب نیست. دو تابع $(x-۲)^۲$ و $(x+۲)^۲$ را در نظر بگیرید. در این حالت تابع تشکیل شده ی r محدب نخواهد بود، نقاط $x = ۲, x = -۲$ را بررسی کنید.
- محدب نیست. برای مثال دو تابع $f(x) = x$ و $g(x) = -x$ را در نظر بگیرید، جفت آنها محدب هستند ولی حاصل ضرب آنها، $s(x) = -x^۲$ محدب نیست.

(ب) از تعریف محدب بودن مجموعه استفاده می کنیم.

convex set: $\forall p_۱, p_۲ \in S,\ \forall \lambda \in [۰, ۱],\ \lambda p_۱ + (۱-\lambda)p_۲ \in S$

اگر ضرب اسکالر و جمع بر روی این زوج های مرتب به این صورت تعریف شود:

$$p_۱ = (x_۱, t_۱), \quad p_۲ = (x_۲, t_۲), \quad p_۱ + p_۲ = (x_۱ + x_۲, t_۱ + t_۲), \quad \lambda * p_۱ = (\lambda * x_۱, \lambda * t_۱)$$

$$\lambda p_۱ = (\lambda x_۱, \lambda t_۱), (۱-\lambda)p_۲ = ((۱-\lambda)x_۲, (۱-\lambda)t_۲)$$

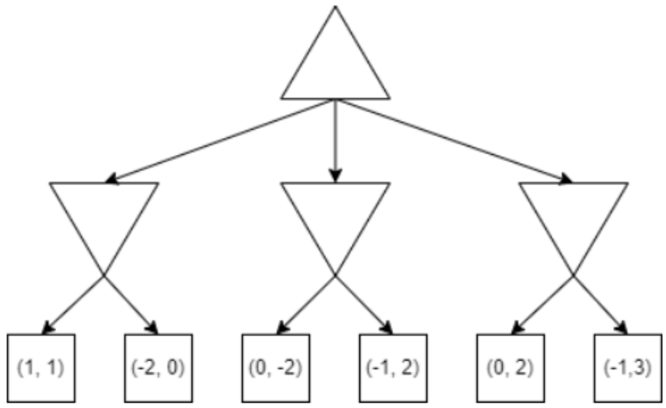
$$\lambda p_۱ + (۱-\lambda)p_۲ = (\lambda x_۱ + (۱-\lambda)x_۲, \lambda t_۱ + (۱-\lambda)t_۲)$$

triangle inequality: $\|\lambda x_۱ + (۱-\lambda)x_۲\| \leq \|\lambda x_۱\| + \|(۱-\lambda)x_۲\|$

$$\leq \lambda\|x_۱\| + (۱-\lambda)\|x_۲\| \xrightarrow{\|x_i\| \leq t_i} \leq \lambda t_۱ + (۱-\lambda)t_۲$$

پس این مجموعه یک مجموعه ی محدب است.

۵. (۲۰ نمره، درجه سختی ۸) فرض کنید در حال بررسی کردن یک بازی non zero-sum هستیم. درخت بازی مورد بررسی به صورت زیر می باشد: (توجه کنید که مثلث‌های رو به بالا و پایین نشان دهنده دو بازیکن متفاوت هستند وگرنه بدیهتا در چنین بازی‌هایی مشخص کردن دقیق یک بازیکن maximizer و یک بازیکن minimzier چندان معنی‌دار نیست چون شرط بازی‌های zero-sum یعنی $U_A(s)+U_B(s) = ۰$ دیگر برقرار نیست و بنابراین هر بازیکن به دنبال maximize کردن امتیاز خود خواهد بود.)



شکل ۲

هر جفت عدد در برگ‌ها به ترتیب امتیاز بازیکن اول و دوم را نمایش می‌دهد. بازیکن اول را A و بازیکن دوم را B می‌نامیم و بنابراین هر جفت عدد به فرمت (U_A, U_B) می‌باشد.

الف) مقادیر هر راس در درخت بازی مورد نظر را تکمیل نمایید.

ب) به طور خلاصه توضیح دهید چرا روش alpha-beta pruning در تعریف عام از بازی های non zero-sum قابل استفاده نمی‌باشد.

راهنمایی: برای مثال خود می‌توانید حالتی که شرط $U_A(s) = U_B(s)$ برای همه برگ‌ها برقرار باشد را مورد بررسی قرار دهید.

در minimax می‌دانیم که مقدار محاسبه شده برای ریشه (که فرض می‌کنیم بازیکن maxmizer باشد.) اصطلاحا یک مقدار worst-case می‌باشد؛ به این معنا که اگر بازیکن minimizer بهینه ترین عمل ممکن را انتخاب نکند نتیجه امتیاز maximizer هرگز بدتر نخواهد شد.

ج) آیا می‌توان گفت که برای یک بازی non zero-sum نیز مقدار محاسبه شده برای ریشه مشابه توضیحات داده شده worst-case می‌باشد؟ به طور خلاصه توضیح دهید.

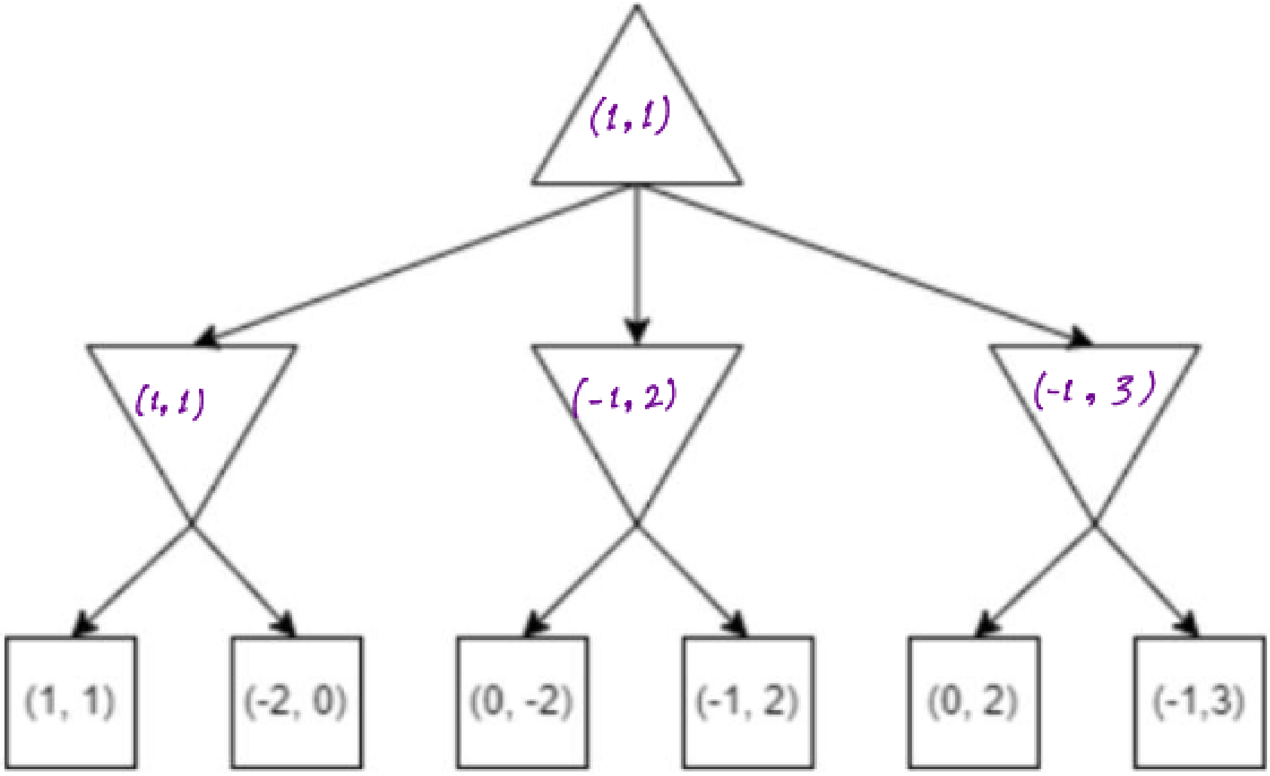
اکنون فرض کنید که بازی تقریبا zero-sum باشد به این معنی که شرط $\mid U_A(s) + U_B(s) \mid \leq \epsilon$ برای تمامی برگ های آن به ازای یک مقدار ϵ مشخص برابر برقرار باشد. مثلا درخت بازی ای که در ابتدای سوال رسم شده است برای مقدار $\epsilon = ۲$ یک بازی nearly zero-sum می‌باشد.

د) در یک بازی nearly zero-sum امکان هرس کردن وجود دارد. با در نظر گرفتن مقدار $\epsilon = ۲$ و تعمیم دادن alpha-beta pruning به بازی کنونی، راس هایی که در طی فرایند هرس کردن خط میخورند را مشخص کنید و توضیحی مختصر درباره الگوریتم در این حالت خاص بدهید. (فرض کنید فرایند هرس کردن به صورت استاندارد آن یعنی از چپ به راست و depth-first انجام می‌شود.)

ه) یک شرط عمومی بیان کنید که تحت آن فرزند یک راس S می‌تواند هرس شود. شرط شما باید با درنظر گرفتن متغیرهایی چون $U_A(S)$ یا $U_B(S)$ برای راس S مربوطه، مقدار ϵ، α و ... بیان شود.

و) در یک بازی nearly zero-sum چه شرطی روی حداقل مقدار امتیازی که ممکن است توسط بازیکن اول کسب شود (برحسب U_A ریشه و ϵ) وجود دارد؟

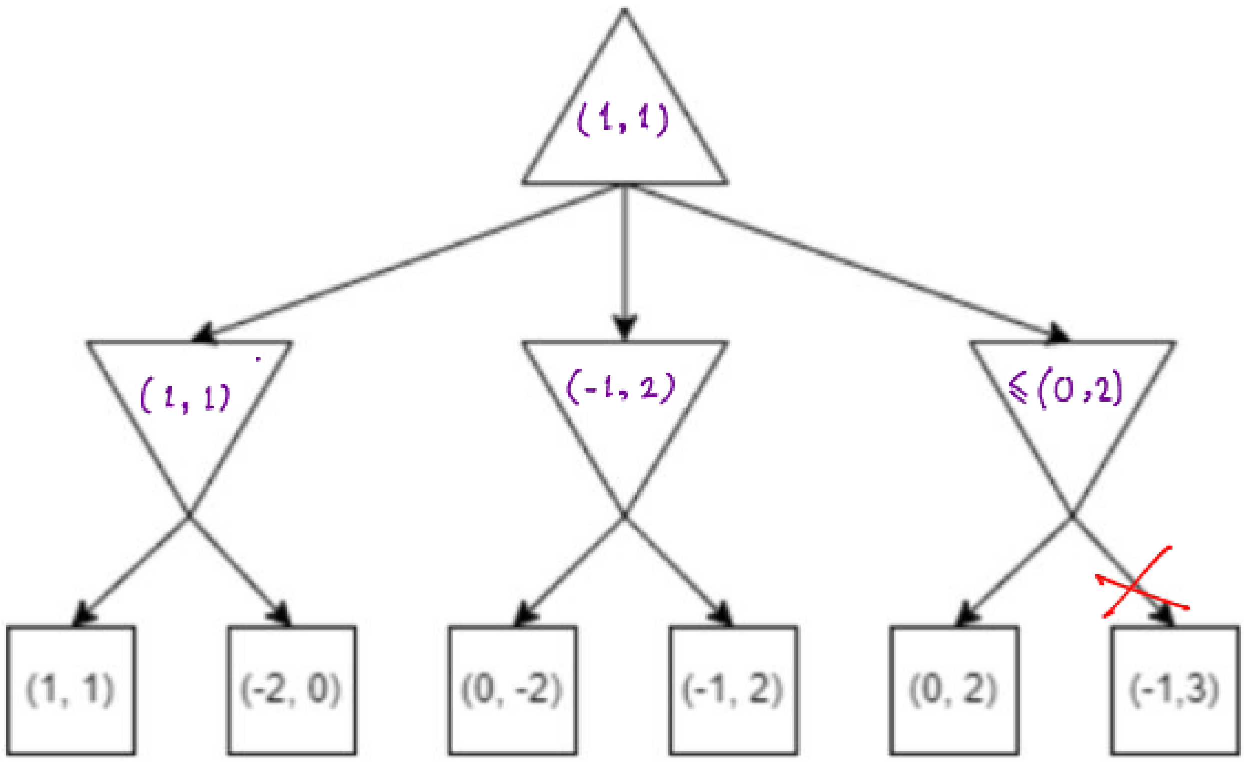
(آ) .



(ب) به این دلیل که این روش بر این منطق کار می کرد که اگر بازیکن minimizer یک مقدار را ببیند قطعا مسیری که انتخاب می کند کمتر مساری آن عدد است. ولی اینجا همچنین شرایطی برقرار نیست. همچنین استدلالی مشابه برای maximizer.

(ج) خیر، زیرا ممکن است حالتی در بررسی حالت ها برای بازیکن دوم حذف شده باشد که ما مقدار کمتری داشته بود ولی برای بازیکن دوم نیز بهترین گزینه نبود. و گزینه ی بهینه ی بازیکن دوم از قضا بهترین گزینه ی ما نیز هست. حال اگر ما این مسیر را انتخاب کنیم، بازیکن دوم با بهینه بازی نکردن می تواند مارا نیز به امتیاز کمتری برساند.

(د) .



در این حالت می توانیم اینگونه فکر کنیم که اگر x امتیاز بازیکن دوم در یک استیت باشد، ما حداکثر $-x+\epsilon$ امتیاز کسب خواهیم کرد از این مسیر و اگر مسیر های بهتری از این امتیاز حداکثری مشاهده شده باشند، prune را انجام می دهیم.

(ه) اگر در مسیر بررسی راس ها یک پارامتر α_S نگه داریم به این صورت که مقدار آن برابر با حداکثر امتیاز کسب شده برای راس S است که با بررسی بچه هایش آپدیت می شود. شرطی که باعث prune شدن شاخه ها می شود به این صورت است:

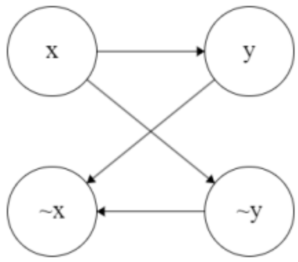
$$\alpha_s \geq -U_B + \epsilon$$

با برقرار این شرط دیگر به بررسی شاخه ی مربوطه ادامه نمی دهیم.

(و) در این حالت، تضمین می شود که حتی اگر بازیکن دوم بهینه بازی نکند، شرایط ما از یک حدی بدتر نخواهد شد. آن حد نیز برابر است با:

$$U_A - ۲ \times \epsilon$$

۶. (- نمره، درجه سختی ؟) (**سوال امتیازی**) می‌دانیم که در حالت کلی پیچیدگی زمانی حل مسئله CSP از اردر نمایی‌است. در این سوال میخواهیم که حالت خاصی از این مسئله به اسم 2-SAT را در اردر خطی حل کنیم.
در این حالت خاص تمام متغیرها باینری (با دامنه ۰ یا ۱) هستند. همچنین تمامی قیود مسئله دوتایی و به شکل $a \lor b$ هستند. یعنی مثلا هیچ قیدی به شکل $a \lor b \lor c$ وجود ندارد.
برای حل ابتدا گرافی جهت دار می‌سازیم که به ازای هر متغیر مثل a دو راس متناظر a و $\neg a$ را قرار می‌دهیم.
برای نمایش قید $p \lor q$ دو یالِ $\neg p \to q$ و $\neg q \to p$ را به گراف اضافه می‌کنیم. درواقع این دو یال به ترتیب معادل این هستند که اگر که گزاره p را False در نظر بگیریم، آنگاه حتما q باید True باشد. و اگر q را False در نظر بگیریم، آنگاه حتما p باید True باشد. به عبارتی از هم‌ارزی $p \lor q \equiv (\neg p \to q) \ \lor (\neg q \to p)$ استفاده شده است.
به عنوان مثال اگر قیدهای مسئله به شکل $(\neg x \lor y) \land (\neg y \lor \neg x)$ باشد، آنگاه گراف متناظر آن به شکل زیر خواهد بود:



شکل ۳

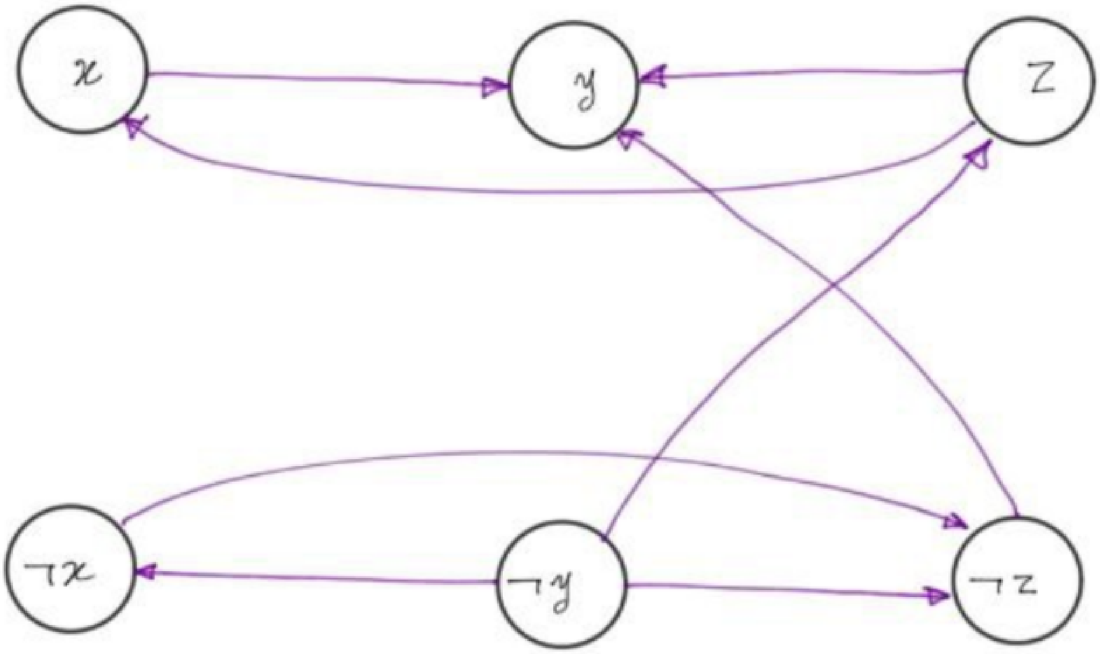
الف) ابتدا مسئله را به صورت یک مسئله CSP بیان کرده و سپس گراف مدنظر را برای مسئله نمونه با قیدهای $(\neg x \lor y) \land (\neg y \lor z) \land (x \lor \neg z) \land (y \lor z)$ رسم کنید و یک جواب برای آن بنویسید.

ب) ادعا میکنیم یک مسئله 2-SAT جواب خواهد داشت اگر و تنها اگر هیچ یک از مؤلفه های قویا همبند این گراف به طور همزمان شاملِ یک متغیر و نقیض آن نباشد. این ادعا را اثبات کنید. (مؤلفه قویا همبند: زیرمجموعه‌ای از رئوس گراف که برای هر جفت راس آن مثل x و y مسیری جهتدار از x به y و برعکس وجود دارد.)

ج) با فرض اینکه در مسئله شرط بخش ب برقرار است (یعنی حتما مقداردهی صحیحی دارد)، یک روش از $O(n + m)$ برای یافتن یک مقداردهی صحیح ارائه کنید. که در آن n تعداد متغیرهاست و m تعداد قیود. (راهنمایی: به مرتب‌سازی توپولوژیک مولفه‌ها فکر کنید.)

(آ)

$$
\begin{aligned}
&x, \quad y, \quad z \\
&D_x = \{۰, ۱\} \quad D_y = \{۰, ۱\} \quad D_x = \{۰, ۱\} \\
&f_۱(x, y) = \neg x \lor y \\
&f_۲(y, z) = \neg y \lor z \\
&f_۳(x, z) = x \lor \neg z \\
&f_۴(y, z) = y \lor z
\end{aligned}
$$



(ب) در شرایط ذکر شده، از گراف به دست آمده می توان این نتیجه گیری را کرد که اگر مسیری از u به v وجود داشته باشد، درست بودن شرط $u \to v$ الزامیست.(طبق انتقالیت یا transitivity)
پس اگر در یک مولفه ی قویا همبند هم x و هم ۴$\neg x$ حضور داشته باشند، به این گزاره می رسیم که $x \to \neg x, \neg x \to x$ که هیچگاه براورده نخواهد شد.

(ج) ابتدا الگوریتمی برای حل این مسئله ارائه می دهیم و بعد نشان می دهیم که در زمان خطی حل می شود.
الگوریتم به این صورت است:

۱. مولفه های قویا همبند را مشخص می کنیم.
۲. پایه ی ترتیب توپولوژیکی را پیدا می کنیم.
۳. شروع به حرکت می کنیم و متغیر ها را مشاهده می کنیم، از هر متغیر، اگر خود متغیر مشاهده شد، به آن ۰ نسبت می دهیم ولی اگر ابتدا نات آن متغیر مشاهده شد، به آن ۱ نسبت می دهیم.

ابتدا نشان می دهیم که این الگوریتم درست است، می دانیم مولفه های قویا همبند این گراف روابط بین متغیر ها را مشخص می کنند، همچنین از خصوصیات این گراف می توان اینگونه گفت که اگر بین u و v یال باشد حتما بین $\neg v$ و $\neg u$ نیز یال هست. حال از پایین ترین این مولفه ها را بر اساس ترتیب توپولوژیکی شروع کرده و مقادیر متغیر ها را مشخص می کنیم. اگر در این ترتیب x قبل از نات خود مشاهده شد به معنی $x \to \neg x$ است یا اینکه گراف ناهمبند است که در هردو حالت اگر ما این شرط را برآورده کنیم، کار درستی است. برای اثبات این قضیه، در حالتی که واقعا مسیری از متغیر به ناتش وجود داشته باشد که باید این شرط برقرار شود و در حالتی که گراف ناهمبند است، فرض کنید که یک متغیر y ای وجود دارد که مقدار نادرست به آن نسبت داده شده و از $\neg x$ به آن یال وجود دارد که با فرض کردن ۰ بودن x دچار مشکل می شویم.
چون از $\neg x$ به y مسیر است پس از $\neg y$ نیز به x مسیر بوده، از آنجایی که مولفه ی همبند x قبل از مولفه ی همبند $\neg x$ رویت شده پس $\neg y$ نیز قبل خود y رویت شده و y باید مقدار ۱ می گرفت که با فرض اولیه در تضاد است پس با اینکار مشکلی پیش نمی آید و الگوریتم درست به متغیر ها مقدار می دهد.
برای اردر زمانی نیز داریم:

۱. با استفاده از الگوریتم کوساراجو در اردر مجموع تعداد یال ها و راس ها
O(n + m)
۲. در اردر تعداد مولفه ها
O(n)
۳. حرکت کردن روی کل گراف
O(n)

پس در کل از O(n + m) است.