Question no.1

Monday, April 8, 2024 12:42 AM

سیدکیهان هدایی - 401106696

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.

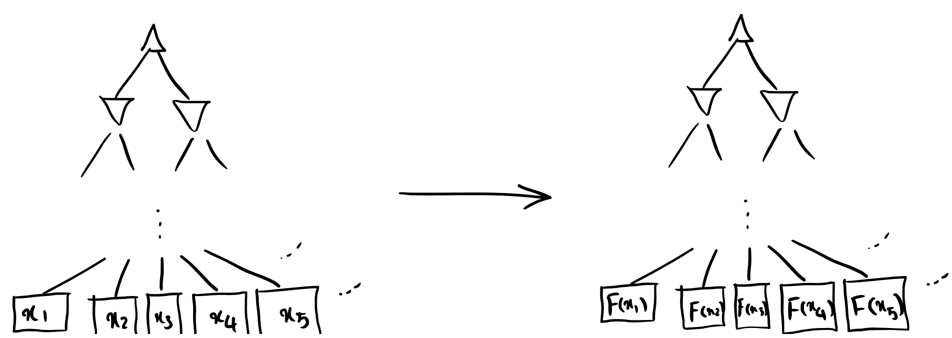
ه) اعمال تابع F روی برگ‌های یک درخت minimax برای این بازی برگ‌هایی که توسط alpha-beta pruning هرس می شوند را تغییر نمی دهد.

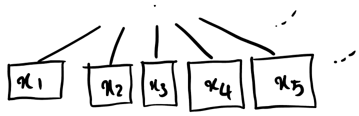
الف) نادرست. در بدترین حالت، به ازای هر رأس یکبار الگوریتم backtracking اجرا می‌شود. با توجه به اینکه از arc-consistency استفاده می‌کنیم می‌توان فهمید که هر بار عمل backtrack در بهترین حالت از $O(n^2d^2)$ است.

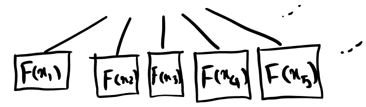
ب) نادرست. ابتدا یک رأس درخت را به عنوان ریشه انتخاب می‌کنیم و به آن یک رنگی را نسبت می‌دهیم ($O(1)$).
در مرحله‌ی دوم، topological sort این درخت را به دست آورده و از آخر ترتیب خطی گراف شروع کرده و تابع <u>remove inconsistent values</u> را روی رأسی که در آن مرحله روی آن هستیم و پدر آن رأس اعمال می‌کنیم تا به رأس ریشه برسیم ($O(nd^2)$).
در مرحله‌ی سوم از ابتدای گراف شروع کرده و به رئوس گراف، رنگ نسبت می‌دهیم ($O(n)$).
بنابراین time complexity این الگوریتم برابر با $O(nd^2+n)$ است که برحسب $\underline{n}$ (d را ثابت فرض کنیم) برابر با $O(n)$ خواهد بود.

ج) نادرست. الگوریتم α,β pruning تنها باعث افزایش سرعت الگوریتم می‌شود (با مقایسه‌ی مقدار node ها) و هیچ تغییری در مقدار خروجی نمی‌دهد.

د) درست.
با اعمال تابع F بر روی برگ ها، ترتیب برگ ها تغییری نمی کند (یعنی اگر برگی از یک برگ دیگر بزرگتر یا کوچکتر بوده، پس از اعمال تابع F بر روی آنها، همچنان همان حالت را حفظ می‌کند). حال می‌خواهیم نشان دهیم که این ترتیب همچنان در طبقات بالا ترجیح برقرار خواهد بود. فرض کنید که رأس پدر، رأس <u>max</u> باشد. بنابراین داریم:







فرض کنید جواب یک رأس در طبقهٔ k قبل از اعمال تابع F، برابر با x_i باشد (یعنی x_i از بقیهٔ بچه‌های این رأس بزرگتر بوده است).
با اعمال تابع F روی این برگها، همچنان $F(x_i)$ از بقیهٔ بچه‌ها بزرگتر خواهد بود (چون F اکیدا صعودی است).
بنابراین مقدار جواب تغییر می‌کند ولی مسیر بهینه تغییری نخواهد کرد.
اگر رأس پدر، رأس min باشد هم همانند بخش قبل به این نتیجه می‌رسیم.

ه) درست. با استدلال مشابه قسمت قبل داریم:

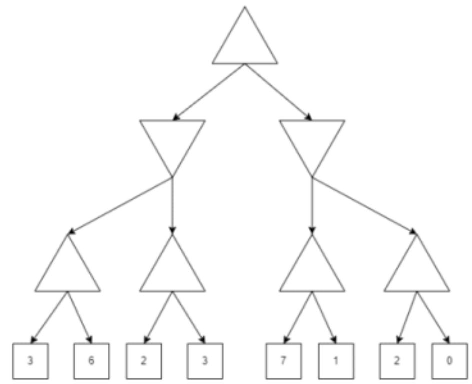
الگوریتم α,β pruning صرفا با مقایسهٔ مقدار رأس‌ها آن‌ها را هرس می‌کند و با توجه به اینکه مقدار رئوس بعد از اعمال تابع F روی برگها همچنان حالت مشابهی را نسبت به همدیگر حفظ می‌کنند (اگر رأس a از b بزرگتر بوده، $F(a)$ هم از $F(b)$ بزرگتر خواهد بود)، تغییری در الگوریتم α,β pruning ایجاد نمی‌شود.

Question no.2

Monday, April 8, 2024 12:42 AM

سیدکیهان هدایتی - 401106696

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



شکل ۱

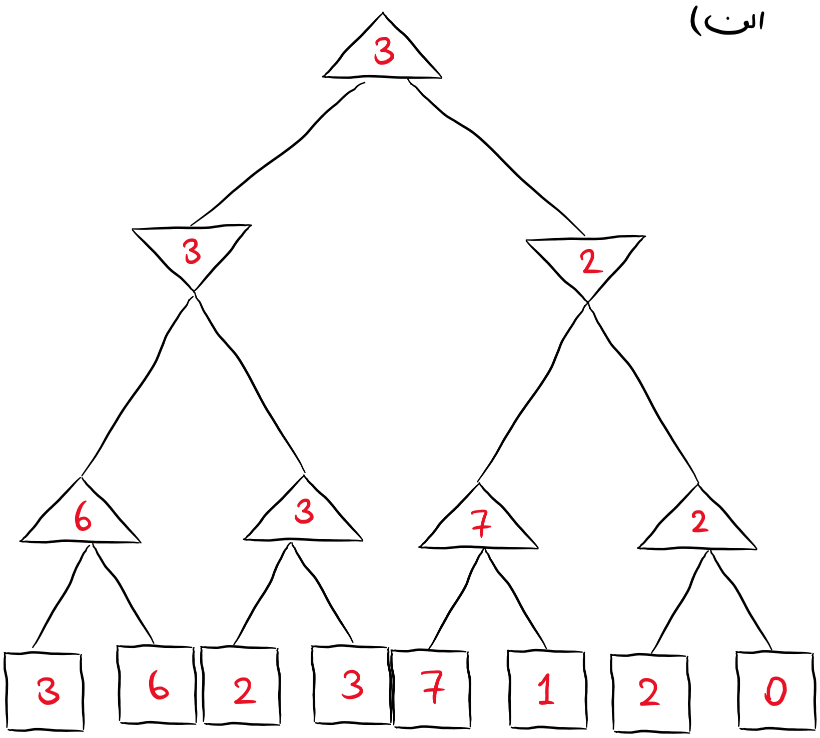
الف) درخت minimax بازی مورد نظر را کامل نمایید.

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.

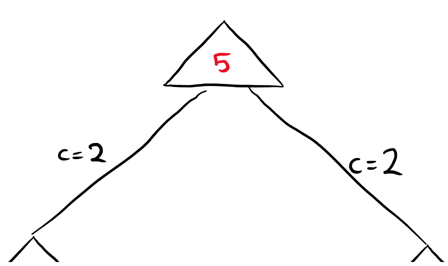
ب) با در نظر گرفتن فرض $c = ۲$ آیا برای بازیکن اول به صرفه خواهد بود که از این قابلیت ویژه استفاده کند؟ درخت بازی را مجدد رسم کرده و کامل نمایید. اگر پاسخ سوال قبل مثبت بود، نقطه ای در درخت که برای بازیکن اول بهینه است از قابلیت ویژه خود استفاده کند را نیز مشخص نمایید.

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

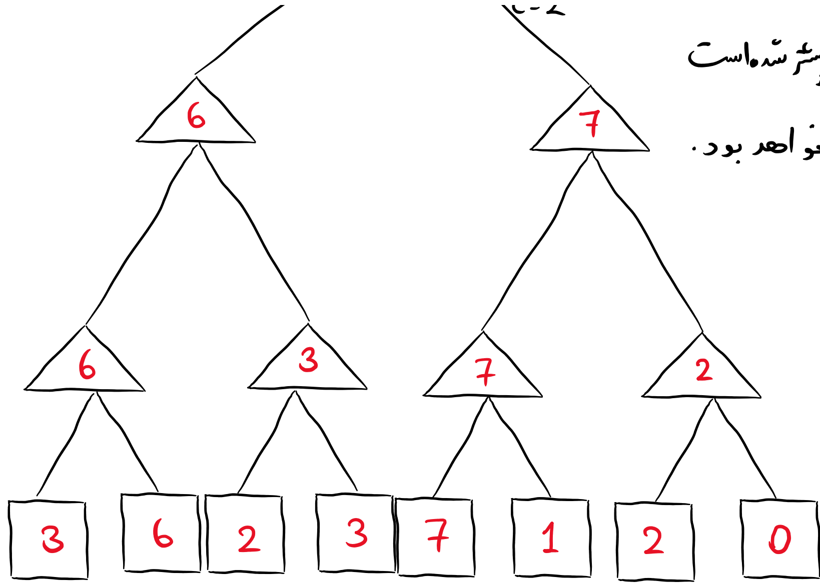
الف)



ب) زمانی که حرکت رقیب را تحت کنترل خود در می آوریم، می توانیم آن حرکت را برای خود بهینه کنیم (به جای اینکه گره، گره‌ی min باشد، گره‌ی max باشد). بنابراین درخت minimax به صورت زیر خواهد بود:

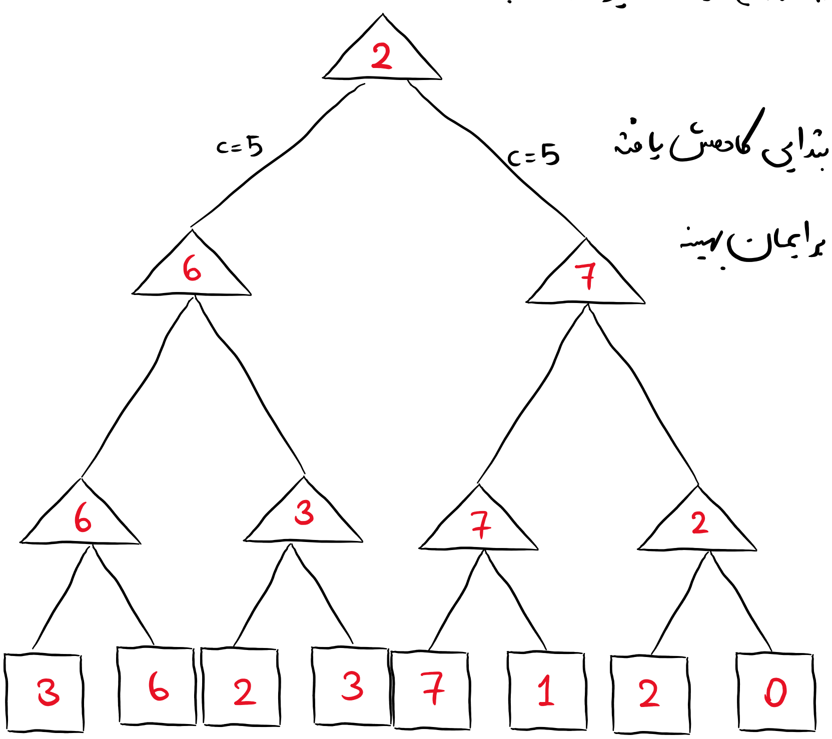


باتوجه به اینکه utility گره‌ی ریشه از حالتی که در ابتدا داشتیم، بیشتر شده است



با توجه به اینکه utility گره‌ی ریشه از حالتی که در ابتدا داشتیم، بیشتر شده است
نتیجه می‌گیریم که استفاده از این قابلیت ویژه برایمان بهینه خواهد بود.

ج) اگر از این قابلیت استفاده کنیم، در این حالت درخت minimax به صورت زیر خواهد بود:



با توجه به اینکه مقدار utility در گره‌ی ریشه نسبت به حالت ابتدایی کاهش یافته
نتیجه می‌گیریم که استفاده از این قابلیت ویژه با چنین هزینه‌ای برایمان بهینه نخواهد بود.

Question no.3

Monday, April 8, 2024 12:43 AM

سیدکیهان هدایی - 401106696

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.

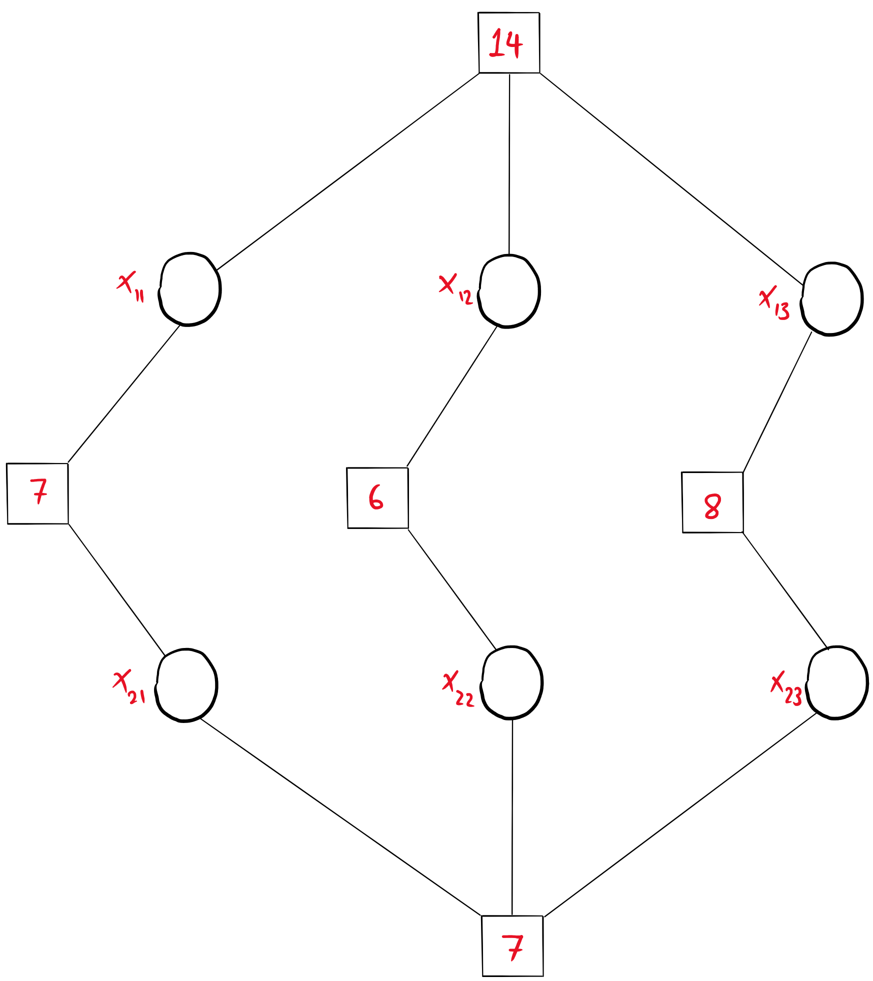
۴.۶	۵.۷	۳.۷	۱۴
۱.۵	۱.۶	۳.۷	۷
۷	۶	۸	

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.

الف) قیود مسئله را بیان می‌کنیم:

$$\begin{cases} x_{11} + x_{21} = 7 \\ x_{12} + x_{22} = 6 \\ x_{13} + x_{23} = 8 \\ x_{11} + x_{12} + x_{13} = 14 \\ x_{21} + x_{22} + x_{23} = 7 \\ (x_{11}=4 \vee x_{11}=5) \wedge (x_{12}=5 \vee x_{12}=6) \wedge (x_{13}=3 \vee x_{11}=4) \wedge (x_{21}=1 \vee x_{21}=2) \wedge (x_{22}=1 \vee x_{22}=2) \wedge (x_{23}=3 \vee x_{23}=4) \end{cases}$$

ب)



ج)

از MRV استفاده می‌کنیم و با توجه به اینکه تعداد مقادیری که می‌توانیم به هر گره نسبت دهیم برابر با 2 است از Degree استفاده می‌کنیم. همچنین درجه همه رئوس باهم برابر است؛ بنابراین یک رأس را به صورت تصادفی انتخاب می‌کنیم و به آن مقدار می‌دهیم:

1) $X_{12} = 5$ $\xleftarrow[\text{به طور تصادفی انتخاب می‌کنیم}]{\text{MRV و Degree گره‌ها باهم برابرند}}$ 2) $X_{21} = 2$ $\xleftarrow[\text{به طور تصادفی انتخاب می‌کنیم}]{\text{MRV و Degree گره‌ها باهم برابرند}}$ 3) $X_{13} = 4$

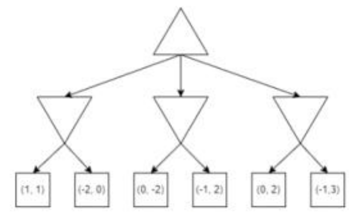
$\xleftarrow[\text{به طور تصادفی انتخاب می‌کنیم}]{\text{MRV و Degree گره‌ها باهم برابرند}}$ 4) $X_{11} = 4$ $\xleftarrow[\text{برقرار نخواهد بود}]{\text{قید مربوط به ستون اول}}$ 5) $X_{11} = 5$ $\xleftarrow[\text{به طور تصادفی انتخاب می‌کنیم}]{\text{MRV و Degree گره‌ها باهم برابرند}}$ 6) $X_{22} = 1$

$\xleftarrow[\text{به طور تصادفی انتخاب می‌کنیم}]{\text{MRV و Degree گره‌ها باهم برابرند}}$ 7) $X_{23} = 4$

5	5	4	14
2	1	4	7
7	6	8	

سیدکیهان هدایی - 401106696

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.

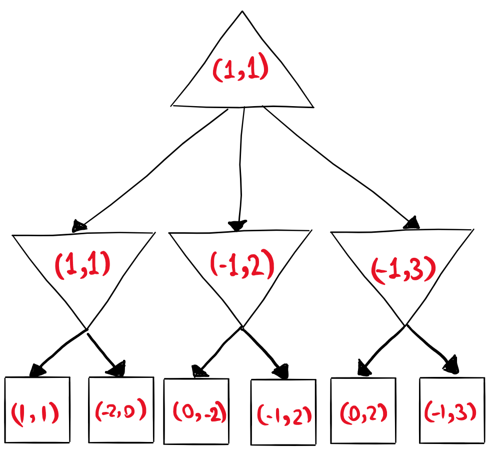
اکنون فرض کنید که بازی تقریبا zero-sum باشد به این معنی که شرط $|\ U_A(s)+U_B(s)\ |\leq\epsilon$ برای تمامی برگ های آن به ازای یک مقدار ϵ مشخص برابر برقرار باشد. مثلا درخت بازی ای که در ابتدای سوال رسم شده است برای مقدار $\epsilon=۲$ یک بازی nearly zero-sum می‌باشد.

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؟

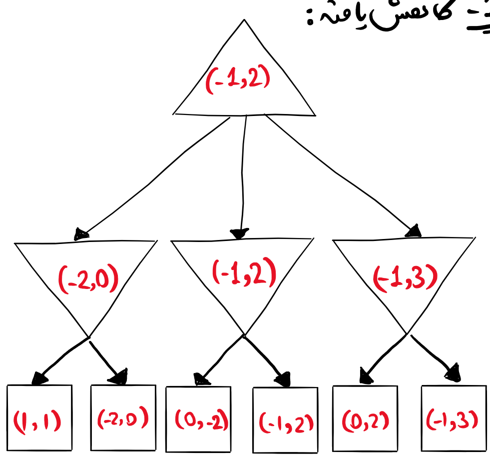
الف)

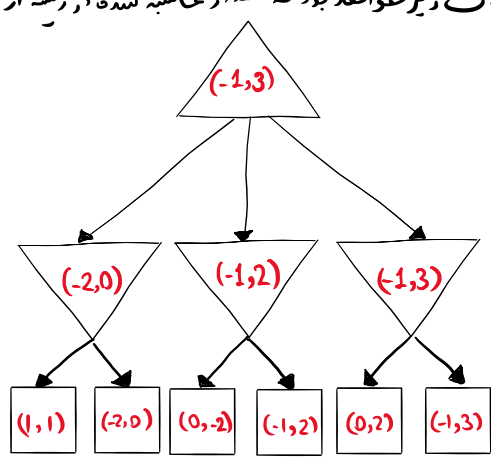


ب) مقادیری که بازیکن های A و B درحال تلاش برای maximize کردنشان هستند، مستقل هستند. بنابراین دیگر موقعیتی پیش نمی آید که یک بازیکن، به ضرر بازیکن دیگر در یک شاخه عمل کند.

برای مثال، در شرایطی که $U_A=U_B$ باشد، مسئله به مسئله‌ی پیداکردن بزرگترین برگ تغییر می کند که می تواند در هرجایی از درخت رخ دهد.

ج) خیر، مقدار محاسبه شده برای ریشه می تواند بدتر باشد. برای مثال، اگر بازیکن B به جای انتخاب حرکت (1,1)، (-2,0) را انتخاب کند، درخت بازی مشابه یکی از حالات زیر خواهد بود که مقدار محاسبه شده در ریشه از 1 به -1 کاهش یافته:





د)

در ابتدا مقدار راس سمت چپ را به دست می آوریم که (1,1) خواهد بود. بعد از آن به سراغ محاسبه‌ی راس وسطی می رویم. ابتدا برگ سمت چپ ((0, -2)) را می بینیم. مقدار U_A در این برگ 0 است و با توجه به شرط $|U_A+U_B|\leq 2$ می بینیم که باید $U_A\geq 0$ باشد تا این

چپ ((0,-2)) را می‌بینیم. مقدار U_A در این برگ 0 است و با توجه به شرط $|U_A+U_B|\le 2$ می‌بینیم که باید $U_A \ge 0$ باشد تا با این برگ رقابت کند. با توجه به اینکه رأس سمت چپ دارای $U_A=1$ است، این برگ را prune می‌کنیم.
حال به سراغ رأس سمت راست می‌رویم. با استدلال مشابه می‌بینیم که در اینجا برگ $(0,2)$ prune می‌شود.

ه)

در حالتی که بازی به صورت zero-sum بود، بازیکن A در حال maximize کردن U_A و بازیکن B در حال minimize کردن U_A بود، که این کار بازیکن B معادل با maximize کردن $-U_A$ است. بنابراین در بازی‌ها zero-sum داریم: $U_B=-U_A$

می‌دانیم که در الگوریتم pruning α, β می‌توان رأسی که تحت تأثیر بازیکن B هست را در صورتی که مقداری کمتر از α داشته باشد، prune کرد. بنابراین اگر شرایط زیر برقرار باشند، رأس مدنظر prune می‌شود:

$$U_A=-U_B<\alpha \rightarrow U_B>-\alpha$$

حال می‌خواهیم مسئلهٔ ε-sum را بررسی کنیم.

$$|U_A+U_B|\le\varepsilon \rightarrow \left.\begin{array}{l} U_A \le \varepsilon - U_B \\ U_A<\alpha\end{array}\right\} \varepsilon-U_B<\alpha \rightarrow U_B>\varepsilon-\alpha$$

و)

می‌دانیم که در صورت بهینه بازی کردن دو بازیکن، امتیاز بازیکن A برابر با U_A و امتیاز بازیکن B برابر با U_B خواهد بود.
اگر بازیکن B بهینه بازی نکند، امتیازش به صورت زیر به دست می‌آید:

$$\exists\, \delta\ge 0:\ U'_B=U_B-\delta$$

$$|U'_A+U'_B|\le\varepsilon \rightarrow -\varepsilon\le U'_A+U'_B\le\varepsilon \rightarrow -\varepsilon\le U'_A+U_B-\delta\le\varepsilon \rightarrow U'_A\ge\delta-U_B-\varepsilon \quad (I)$$

$$|U_A+U_B|\le\varepsilon \rightarrow -\varepsilon\le U_A+U_B\le\varepsilon \rightarrow -\varepsilon-U_A\le U_B\le\varepsilon-U_A \quad (II)$$

$$\xrightarrow[(I),(II)]{} U'_A\ge\delta-\varepsilon-(\varepsilon-U_A)=\delta-\varepsilon-\varepsilon+U_A=U_A-2\varepsilon+\delta$$

$$\rightarrow U'_A\ge U_A-2\varepsilon+\delta,\ \delta\ge 0 \rightarrow U'_A\ge U_A-2\varepsilon$$

Question no.6

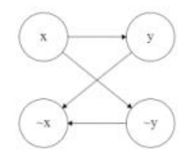
Monday, April 8, 2024 12:43 AM

401106696 - سیدکیهان هدایی

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 $a \vee b$ هستند. یعنی مثلا هیچ قیدی به شکل $a \vee b \vee c$ 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 $p \vee q$ 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 $p \vee q \equiv (\neg p \to q) \vee (\neg q \to p)$ استفاده شده است.

به عنوان مثال اگر قیدهای مسئله به شکل $(\neg x \vee y) \wedge (\neg y \vee \neg x)$ باشد، آنگاه گراف متناظر آن به شکل زیر خواهد بود:



شکل ۳

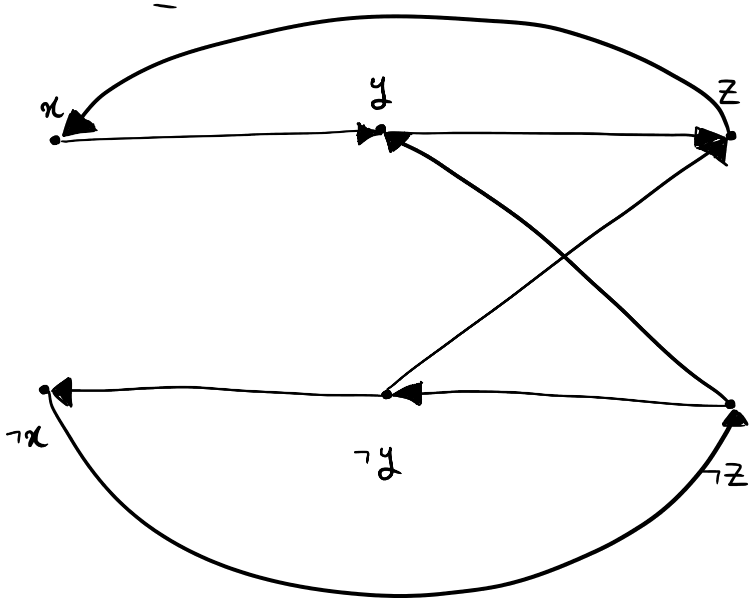
الف) ابتدا مسئله را به صورت یک مسئله CSP بیان کرده و سپس گراف مدنظر را برای مسئله نمونه با قیدهای $(\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (y \vee z)$ رسم کنید و یک جواب برای آن بنویسید.

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 شامل 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.)

الف) فرض کنیم مسئله به صورت $(x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \ldots \wedge (x_{n-1} \vee x_n) = \bigwedge_{i=1}^{n-1} (x_i \vee x_{i+1})$ باشد.
متغیرهای ما به صورت x_i هستند که دامنه‌ی هر کدام از آن ها $\{0,1\}$ می باشد و قیود ما $(x_i \vee x_{i+1})$ ها هستند که همگی باید برقرار و برابر با 1 باشند.

گراف 2-SAT :



یک جواب برای مسئله :

$x = 1, y = 1, z = 1$

ب)

ب)

$\Leftarrow$ اثبات رفت:

فرض می‌کنیم مسئله‌ی 2-SAT برقرار است (یعنی تمامی قیود برقرارند و برابر با 1 هستند). اگر برای برقراری یکی از قیود متغیر x برابر با 1 باشد داریم:

برهان خلف: فرض می‌کنیم که در مؤلفه‌ی همبند گراف از متغیر x به $\neg x$ مسیری وجود دارد. این به این معناست که اگر برای برقراری یکی از قیود، متغیر x برابر با 1 باشد برای برقراری یکی دیگر از قیود متغیر $\neg x$ هم باید برابر با 1 باشد که چنین حالتی ممکن نیست (که یک متغیر و نقیضش با هم برابر باشند).

به تناقض رسیدیم. پس، مسئله‌ی 2-SAT جواب دارد، اگر هیچ کدام از مؤلفه‌های قویا همبند این گراف به طور همزمان شامل یک متغیر و نقیضش نباشند. (I)

$\Rightarrow$ اثبات برگشت:

نشان می‌دهیم که اگر در هیچ‌کدام از مؤلفه‌های همبند گراف، متغیری و نقیضش همزمان وجود نداشته باشند، مسئله‌ی 2-SAT جواب دارد:

جوابی را برای این حالت ارائه می‌دهیم:

مؤلفه‌های قویا همبند گراف را به عنوان رأس در نظر می‌گیریم و topological sort این گراف جدید را به دست می‌آوریم:

$$C_1 \to C_2 \to C_3 \to C_4 \to \cdots \to C_n$$

C_i ها مؤلفه‌ی قویا همبند هستند.

از مؤلفه‌ی اول شروع به گردش می‌کنیم؛ به ازای هر متغیر x در گراف، اگر در این گردش ابتدا x مشاهده شد به آن مقدار 0 را نسبت می‌دهیم ($x=0$) و اگر $\neg x$ ابتدا دیده شد به آن مقدار 1 را نسبت می‌دهیم ($\neg x = 0 \rightsquigarrow x = 1$).

حال می‌خواهیم نشان دهیم که این جواب، یک جواب معقول و مناسب است:

فرض خلف: مسئله‌ی 2-SAT جواب ندارد، یعنی قیدی مانند $(x \vee y)$ وجود دارد که برقرار نیست و برابر با 0 است.

$(x \vee y) = 0 \rightarrow x = y = 0$

i) فرض کنید $idx(x)$ نشان دهندهٔ اندیس گرهٔ x در گراف بالاست.

ii) $x=0$ است. بنابراین طبق فرض x زودتر از $\neg x$ در ترتیب گراف قرار گرفته و در نتیجه $idx(x) < idx(\neg x)$.
به همین ترتیب $idx(y) < idx(\neg y)$.

iii) با توجه به مسئلهٔ 2-SAT، به دلیل وجود قید $(x \vee y)$ در مسئله، در گراف یال‌های $\neg x \rightarrow y$ و $\neg y \rightarrow x$ وجود دارند که نتایج $idx(\neg x) < idx(y)$ و $idx(\neg y) < idx(x)$ را می‌دهند.

$$\left.\begin{array}{l} idx(\neg x) < idx(y) \\ idx(\neg y) < idx(x) \end{array}\right\} \xrightarrow{idx(x) < idx(\neg x)} idx(\neg y) < idx(x) < idx(\neg x) < idx(y) \rightarrow idx(\neg y) < idx(y)$$

تناقض با مورد (ii): $idx(y) < idx(\neg y)$

بنابراین جواب ارائه شده درست است. پس:

اگر هیچ کدام از مؤلفه‌های قویا همبند این گراف به طور همزمان شامل یک متغیر و نقیضش نباشند، مسئلهٔ 2-SAT جواب دارد. (II)

(I), (II) ← مسئلهٔ 2-SAT جواب دارد، اگر و تنها اگر هیچ کدام از مؤلفه‌های قویا همبند این گراف به طور همزمان شامل یک متغیر و نقیضش نباشند.

ج)

الگوریتم به دست آوردن جواب را در بخش قبل بررسی کردیم. در این جا time complexity این الگوریتم را به دست می‌آوریم:

(I) با هزینهٔ $O(m)$ گراف را (با n گره و m یال) می‌سازیم.

(II) به صورت زیر با هزینه‌ی $O(n+m)$ مؤلفه‌های قویاً همبند گراف را پیدا می‌کنیم:

در ابتدا از یک رأس به عنوان ریشه DFS می‌زنیم و رئوس را به ترتیبی که می‌بینیم (starting time) در یک لیست قرار می‌دهیم.

حال به ترتیب مؤلفه‌های موجود در لیست را مشاهده کرده و از آن‌ها روی گراف برعکس (گرافی که جهت یال‌هایش برعکس است) DFS می‌زنیم. همه‌ی رئوسی که در این DFS برعکس دیده می‌شوند، یک مؤلفه‌ی قویاً همبند هستند.

(ارجاع به الگوریتم tarjan's strongly connected components)

(III) حال هر مؤلفه‌ی قویاً همبند را یک رأس در نظر گرفته و topological sort این گراف جدید را در $O(n+m)$ به دست می‌آوریم.

(IV) در این گراف جدید به دست آمده می‌گردیم و به ازای هر مؤلفه‌ی x، اگر x را زودتر دیدیم $x=0$ و اگر $\neg x$ را زودتر دیدیم $x=1$ را نسبت می‌دهیم که این کار در $O(n)$ انجام می‌پذیرد.

$$\text{time complexity} = O(m) + O(n+m) + O(n+m) + O(n) = O(n+m)$$

Question no.4

Monday, April 8, 2024 12:50 AM

۴. (۲۰ نمره، درجه سختی ۷)

الف) با فرض آنکه $f, g : \mathbb{R} \to \mathbb{R}$ محدب‌اند و $t \geq 0$ 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.

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

401106696 - سیدکیان هدایی

الف)

I) f is convex: $f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y)$ for $\forall x, y$ in D_f and $0 \leqslant \theta \leqslant 1$

g is convex: $g(\theta x + (1-\theta)y) \leqslant \theta g(x) + (1-\theta) g(y)$ for $\forall x, y$ in D_g and $0 \leqslant \theta \leqslant 1$

$\rightarrow tg(\theta x + (1-\theta)y) \leqslant t\theta g(x) + t(1-\theta) g(y) \rightarrow tg$ is convex

$$\left.\begin{array}{l} f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y) \\ tg(\theta x + (1-\theta)y) \leqslant t\theta g(x) + t(1-\theta) g(y) \end{array}\right\} f(\theta x + (1-\theta)y) + tg(\theta x + (1-\theta)y) \leqslant \theta f(x) + t\theta g(x) + (1-\theta) f(y) + t(1-\theta) g(y)$$

$\rightarrow (f + tg)(\theta x + (1-\theta)y) \leqslant \theta(f + tg)(x) + (1-\theta)(f + tg)(y) \rightarrow$ $h(x)$ is convex

II) f is convex: $f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y)$ for $\forall x, y$ in D_f and $0 \leqslant \theta \leqslant 1$

g is convex: $g(\theta x + (1-\theta)y) \leqslant \theta g(x) + (1-\theta) g(y)$ for $\forall x, y$ in D_g and $0 \leqslant \theta \leqslant 1$

$$\rightarrow max\{f(\theta x + (1-\theta)y), g(\theta x + (1-\theta)y)\} \leqslant max\{\theta f(x) + (1-\theta) f(y), \theta g(x) + (1-\theta) g(y)\}$$

$$max\{f(\theta x + (1-\theta)y), g(\theta x + (1-\theta)y)\} = \left\{\begin{array}{l} f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y) \leqslant max\{\theta f(x) + (1-\theta) f(y), \theta g(x) + (1-\theta) g(y)\} \\ \\ g(\theta x + (1-\theta)y) \leqslant \theta g(x) + (1-\theta) g(y) \leqslant max\{\theta f(x) + (1-\theta) f(y), \theta g(x) + (1-\theta) g(y)\} \end{array}\right.$$

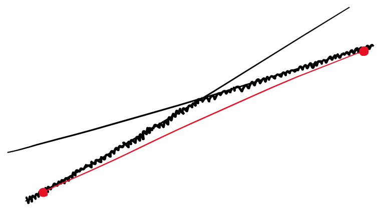
$$\rightarrow max\{f(\theta x + (1-\theta)y), g(\theta x + (1-\theta)y)\} \leqslant max\{\theta f(x), \theta g(x)\} + max\{(1-\theta) f(y), (1-\theta) g(y)\}$$

$\rightarrow$ $k(x)$ is convex

III)

دو معادله‌ی خط را در نظر بگیرید (طبق تعریف تابع محدب، این دو تابع هم محدب اند). فرض کنید این دو خط همدیگر را قطع می‌کند.

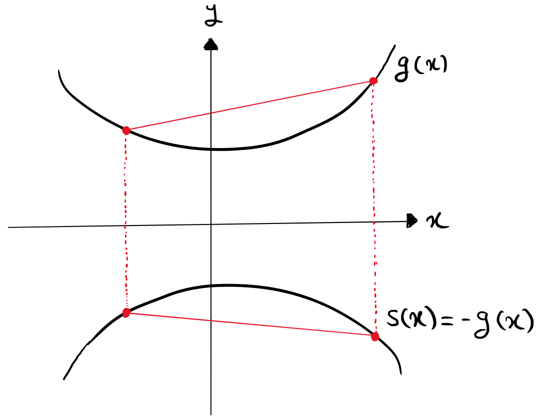
تابع $\min\{f(x),g(x)\}$ را با خط شکسته نشان می‌دهیم:



حال اگر هر دو نقطه‌ای را روی این تابع در نظر بگیریم و به هم وصل کنیم می‌بینیم که خط واصل این دو در زیر این تابع قرار دارد. پس نتیجه می‌گیریم که تابع $r(x)$ محدب نیست.

IV)

فرض کنید تابع $f(x)$ خط ثابت -1 باشد و $g(x)$ هم یک تابع محدب دلخواه. با ضرب این دو تابع در هم، تابع خروجی قرینه‌ی تابع g نسبت به محور xها خواهد بود و در این حالت هر دو نقطه‌ای را روی این تابع در نظر بگیریم و به هم وصل کنیم، خط واصل زیر نمودار قرار بود و در نتیجه تابع $s(x)$ محدب نیست.



ب)

if C is convex $\rightsquigarrow$ $\forall x, y \in C,\ 0 \leqslant \theta \leqslant 1:\ \theta x + (1-\theta) y \in C$

$$\left.\begin{array}{l} (x_1, t_1) \in C \rightsquigarrow \|x_1\| \leqslant t_1 \\ (x_2, t_2) \in C \rightsquigarrow \|x_2\| \leqslant t_2 \end{array}\right\} \theta(x_1, t_1) + (1-\theta)(x_2, t_2) = (\theta x_1 + (1-\theta) x_2,\ \theta t_1 + (1-\theta) t_2)$$

$$\longrightarrow \|\theta x_1 + (1-\theta) x_2\| \overset{\text{triangle inequality}}{\leqslant} \|\theta x_1\| + \|(1-\theta) x_2\| = \theta \|x_1\| + (1-\theta)\|x_2\|$$

$$\left.\begin{array}{l} \|x_1\| \leqslant t_1 \\ \|x_2\| \leqslant t_2 \end{array}\right\} \longrightarrow \|\theta x_1 + (1-\theta) x_2\| \leqslant \theta\|x_1\| + (1-\theta)\|x_2\| \leqslant \theta t_1 + (1-\theta) t_2 \longrightarrow C \text{ is convex}$$