## سوال ۱)

پاسخ:

الف)

متغیرها: $X_1$ و $X_2$ و $X_3$ و $X_4$

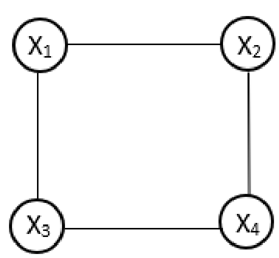
دامنه متغیرها:

$X_1 = X_4 = \{ FALK, FALL, SPEL \}$ و $X_2 = X_3 = \{ APA, ASP, KAP, KUL \}$

محدودیت‌ها:

1) $X_1[2] = X_2[1]$
2) $X_1[4] = X_3[1]$
3) $X_4[2] = X_2[3]$
4) $X_4[4] = X_3[3]$

گراف محدودیت:




ب)

| یال‌های بررسی شده | تغییرات |
|---|---|
| $(X_1, X_2)$ | حذف SPEL از دامنه $X_1$ |
| $(X_1, X_3)$ | حذف FALL از دامنه $X_1$ |
| $(X_2, X_1)$ | حذف KAP و KUL از دامنه $X_2$ |

| | |
|---|---|
| $(X_2, X_4)$ | |
| $(X_3, X_1)$ | حذف APA و ASP از دامنه $X_3$ |
| $(X_3, X_4)$ | حذف KAP از دامنه $X_3$ |
| $(X_4, X_2)$ | |
| $(X_4, X_3)$ | حذف FALK از دامنه $X_4$ |
| $(X_1, X_2)$ | |
| $(X_1, X_3)$ | |
| $(X_2, X_4)$ | |
| $(X_3, X_4)$ | |

دامنه متغیرها بعد از اعمال arc consistency:

$X_1$ = { FALK }, $X_2$ = { APA, ASP }, $X_3$ = { KUL }, $X_4$ = { FALL, SPEL }

ج)

یک پاسخ برای این مساله:

| | | | |
|---|---|---|---|
| F | A | L | K |
| [redacted] | S | [redacted] | U |
| S | P | E | L |

## سوال ۲)

پاسخ:

الف)

2) A = { 2, 3, 4, 5, 6 }

3) B = { 2, 3, 6 }

4) F = { 1, 3, 5 }

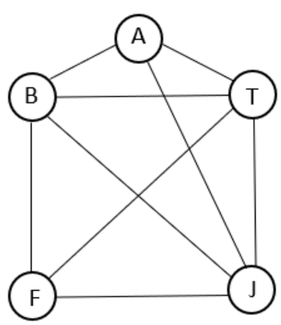
5) J = {1, 2, 3, 4 }

7) T = {1, 2, 3, 4, 5 }

ب)

(J, F) = { (3, 4), (4, 3), (2, 5), (2, 6), (5, 2), (5, 6), (6, 2), (6, 5), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6) }

ج)



د)

F = 1: محدودیت هشتم J = { 1 }

محدودیت دهم J = { }

در نتیجه انتسابی با F = 1 وجود ندارد.

F = 3: محدودیت هشتم J = { 3, 4 }

محدودیت دهم J = { 4 } و T = { 1, 2, 4, 5 } و A = { 2, 3, 5, 6 }

محدودیت اول B = { 2, 6 }

محدودیت ششم T = { 1, 2 }

محدودیت نهم B = { 6 } و A = { 2, 3, 5 }

انتساب‌های ممکن برای (A, B, F, J, T):

(2, 6, 3, 4, 1), (3, 6, 3, 4, 1), (5, 6, 3, 4, 1), (2, 6, 3, 4, 2), (3, 6, 3, 4, 2), (5, 6, 3, 4, 2)

F = 5: محدودیت هشتم J = { 2 }

محدودیت دهم B = { 3, 6 } و A = { 3, 4, 5, 6 }

محدودیت ششم T = { 1 }

محدودیت نهم B = { 6 } و A = { 3, 4, 5 }

انتساب‌های ممکن برای (A, B, F, J, T):

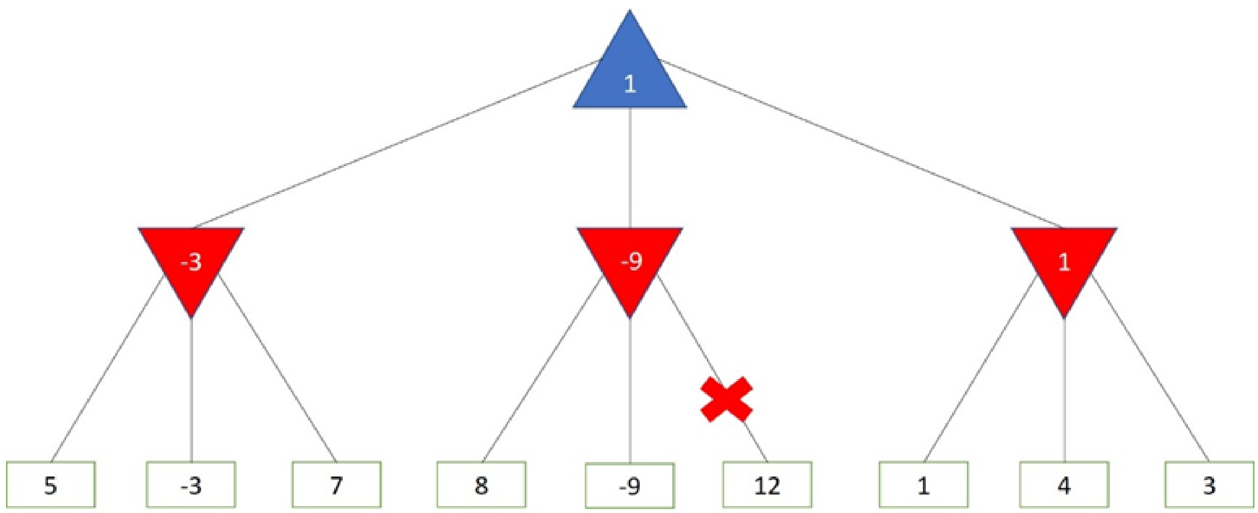
(3, 6, 5, 2, 1), (4, 6, 5, 2, 1), (5, 6, 5, 2, 1)

ه) با توجه به قسمت د با LCV متغیر F مقدار 3 می‌گیرد زیرا کمترین محدودیت را برای سایر متغیرها اعمال می‌کند و با MRV ابتدا متغیر B مقداردهی می‌شود زیرا متغیر با کمترین مقادیر در دامنه شامل B و F می‌باشدکه بر حسب حروف الفبا متغیر B انتخاب می‌شود.

## سوال ۳)

پاسخ:

الف)



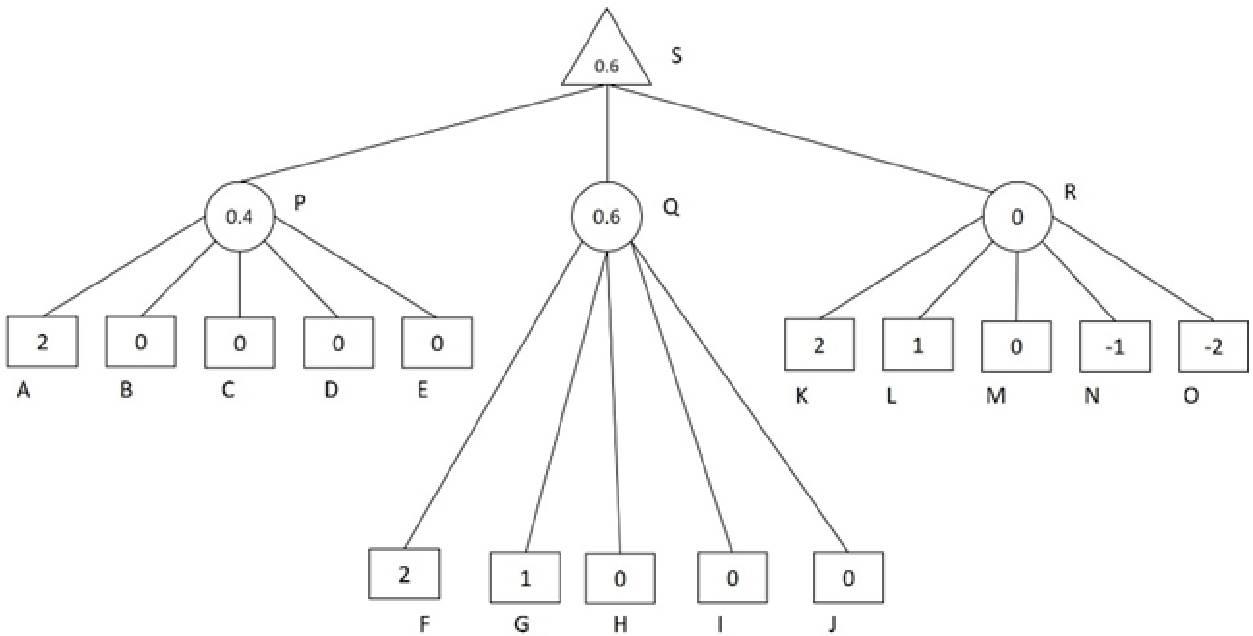

ب) با استفاده از تابع ارزیابی داده شده، مقدار 69 در ریشه قرار می گیرد و باعث می شود که تیم آبی به اشتباه عمل وسط را (به جای عمل سمت راست که در قسمت الف انتخاب کرده بود) انتخاب کند.

ج) برای اینکه بعد از استفاده از تابع ارزیابی عمل انتخاب شده توسط تیم آبی تغییر نکند، باید مقداری را برای برگ ها انتخاب کنیم که در آن تابع ارزیابی اکیدا صعودی باشد. چون تابع ارزیابی از درجه دو است، باید مقدار برگ ها از راس آن بیشتر باشد. چون راس سهمی در نقطه -1 قرار دارد، بازه مورد نظر از 1- تا بینهایت است.

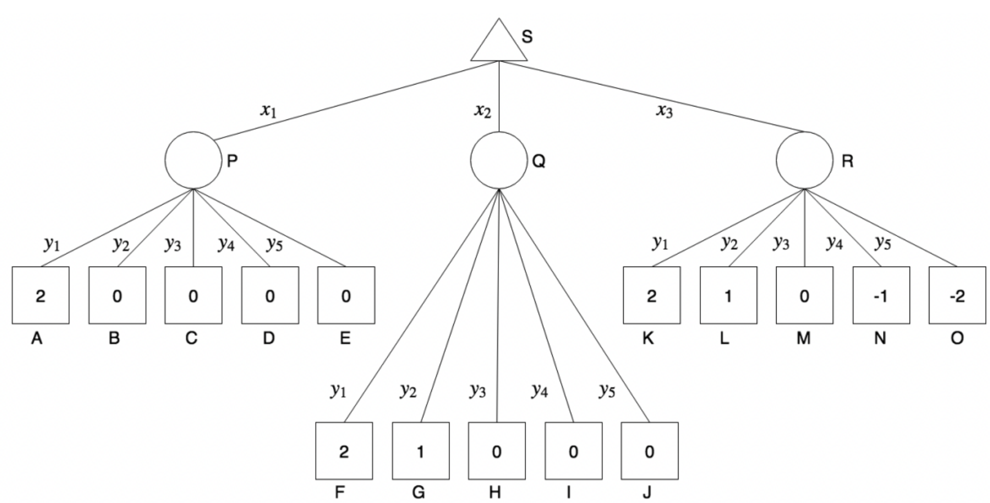
## سوال ۴)

پاسخ:

الف) مطابق با شکل زیر، نود max زیر شاخه وسط را انتخاب می کند:



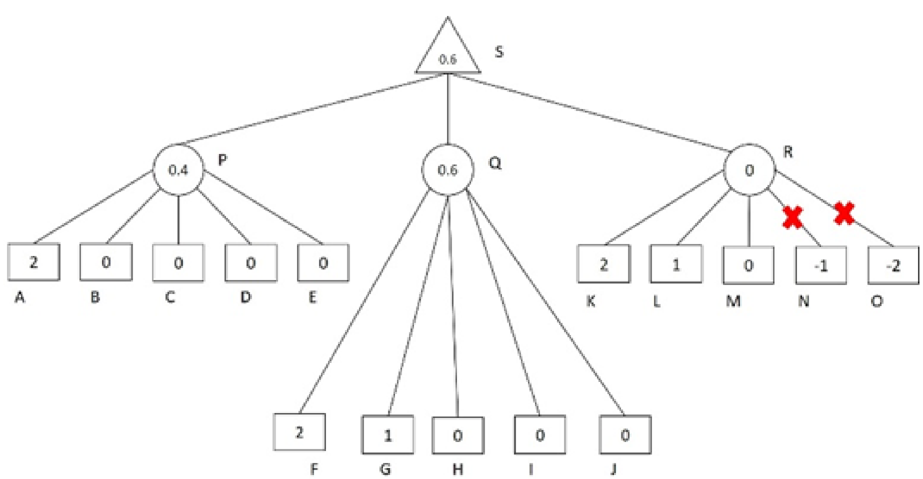

ب) با در نظر گرفتن نوتیشنهای درخت زیر پاسخ را بخوانید



برای هرس فرزندان یک گره شانس در یک درخت expectimax، باید یک مقدار آستانه را در مورد مقدار گره شانس پیگیری کنیم. اگر در نقطه ای در حین جستجو از چپ به راست، متوجه شویم که مقدار گره شانس هرگز بالاتر از آستانه خود نخواهد بود، می توانیم فرزندان باقی مانده گره شانس را هرس کند. ما باید کل زیردرخت سمت چپ را جستجو کند زیرا آستانه ای برای مقایسه با مقدار P نداریم.

وقتی زیردرخت مرکزی را جستجو می‌کنیم، می‌دانیم که گره ماکسیمایزر تنها زمانی اقدام x2 را انجام میدهد که مقدار Q بالاتر از مقدار P باشد که 0.4 است. اگر در جایی متوجه شویم که ارزش Q هرگز بالاتر از 0.4 نخواهد بود، او می تواند بچه های باقی مانده را هرس کند. پس از بررسی نود G متوجه میشویم نودهای I,J,H مقداری کمتر مساوی یک دارند و به این معنی است که مقدار گره Q حداکثر ۱.۲ است. پس از بررسی H متوجه میشویم مقادیر I, J کمتر مساوی صفر است و یعنی مقدار Q حداکثر ۰.۶ است. این اطلاعات برای هرس کردن هیچ یک از زیرشاخه های زیردرخت مرکزی کافی نیست زیرا در هیچ جا مطمئن نشدیم که مقدار Q کمتر مساوی 0.4 است.

وقتی زیر درخت سمت راست را بررسی میکنیم به این نتیجه میرسیم مقدار R هرگز بیشتر از 0.6 نمیشود و آنجا فرزندان باقیمانده را هرس میکنیم. بعد از بررسی گره L میفهمیم گره های M, N , O کمتر مساوی ۱ هستند که یعنی R حداکثر ۱.۲ است.  بعد از بررسی گره M میفهمیم گره های N , O کمتر مساوی 0 هستند که یعنی R حداکثر 0.6 است. در این جا می توانیم گره های N و O را هرس کنیم زیرا آنها فقط می توانند مقدار R را کمتر از مقدار Q کنند.پس داریم:



## سوال ۵)

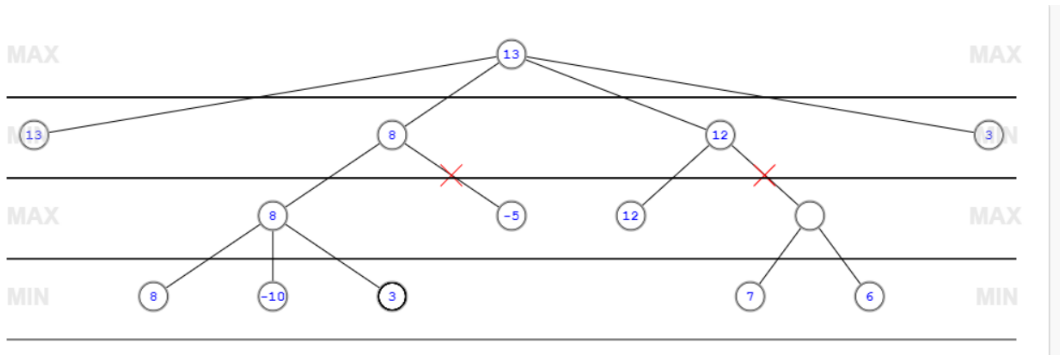
الف)

پاسخ: d

در arc consistency هنگام بررسی یال $X_iX_j$ در صورت حذف هر مقدار غیرمجاز از دامنه $X_i$ مجدد همه یال‌هایی مثل $X_kX_i$ که سر دوم آنها $X_i$ است از نظر سازگاری بررسی می‌شوند. از آنجا که $X_i$ دقیقاً d مقدار در دامنه خود برای حذف شدن دارد پس هر یال حداکثر d بار سازگار می‌شود.

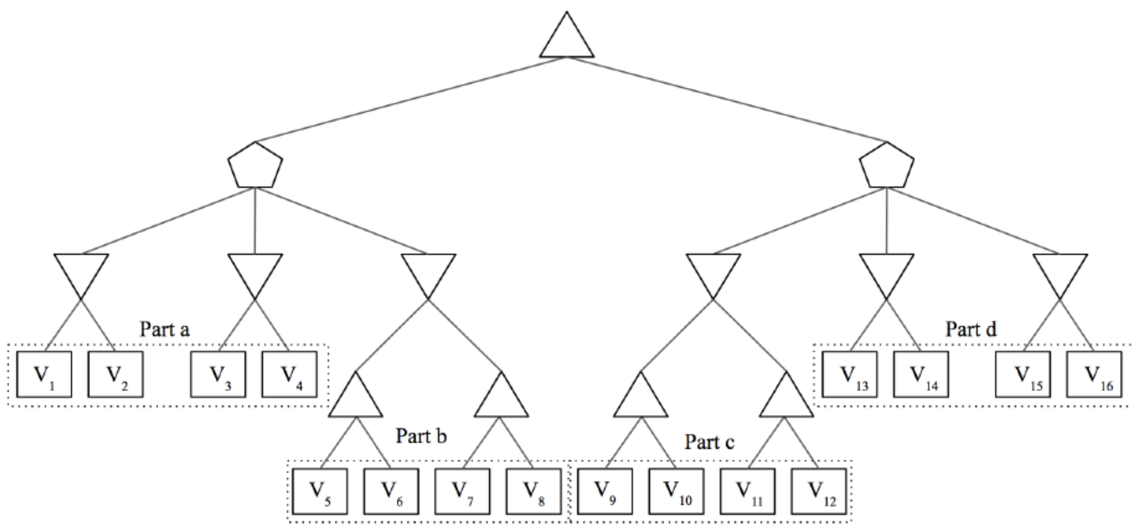
ب)

پاسخ:



## سوال ۶)

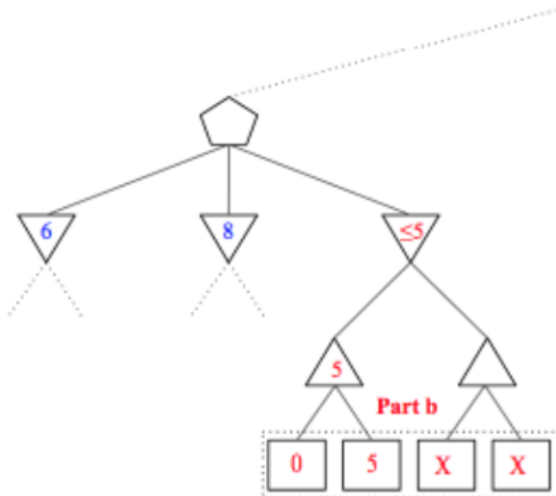
صورت سوال: در درخت minimax زیر عاملی که با پنج ضلعی نشان داده شده است همواره میانه فرزندان خود را انتخاب می کند.در هر کدام از بخش های مشخص شده تعیین کنید کدام یک از ترمینال نودها می توانند هرس شوند.به عبارت دیگر با در نظر گرفتن تمامی حالات هرس کردن درخت، حالتی وجود داشته باشد که ترمینال نود انتخاب شده هرس شود.



پاسخ(سوال اول این لینک را نیز میتوانید ببینید):

بخش a - گره میانه گیر چپی که سه فرزند دارد حداقل نیاز به مقدار دو فرزندش (مینیمم گیرنده ها) دارد تا بتواند میانه را پیدا کند و بنابراین هیچ گره ای از بخش a نمیتواند هرس شود.

بخش b – هرس v7 و v8

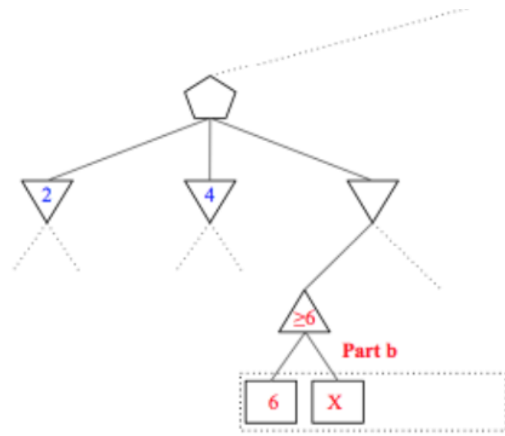



فرض کنید min1 و min2 و min3 مقادیر مینیمم گیرنده ها در این زیر درخت باشند. در مثال بالا نیاز نداریم مقدار دقیق min3 را بدانیم و explore کردن بخشی از v5 تا v8 میتوانیم روی min3 محدودیت گذاشته و با min2 و min1 آنرا مقایسه کنیم.

اگر min3< min(min1,min2) باشد یا min3>=max(min1,min2) میدانیم min3 قطعا میانه نخواهد بود و نیازی به explore کردن ادامه گره های زیر درخت min3 نداریم.

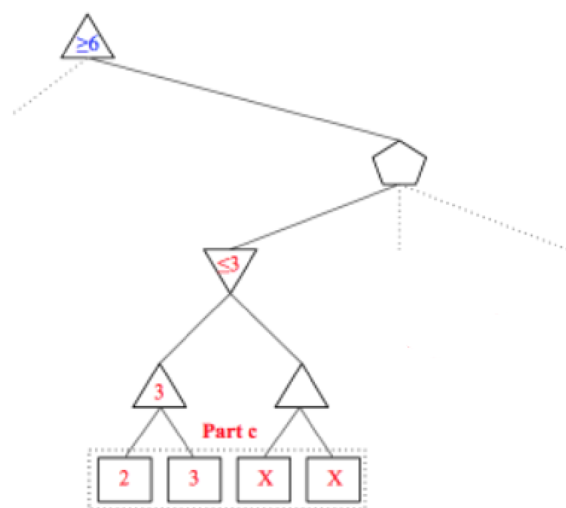
در مثال بالا میبینیم min3 کمتر یا مساوی 5 خواهد بود (یکی از ماکسیمم های زیر min3 ، 5 است پس min3 بین 5 و ماکس دیگری، عدد کمتر را انتخاب میکند پس حاصل min3 کمتر یا مساوی 5 است) و از min2 و min1 که مقادیر 8 و 6 دارند قطعا کمتر است پس میانه نیست پس نیازی به explore کردن گره های v7 و v8 نیست.

بخش b – هرس v6




شکل را در نظر بگیرید. Min3 اگر ماکسیمم چپی را انتخاب کند یعنی عددی بزرگتر یا مساوی v5 که مقدار 6 دارد را انتخاب میکند که از min1 و min2 بزرگتر است و میانه نیست. اگر min3 ماکسیمم راستی را انتخاب کند پس نیازی به explore کردن v6 نداشتیم. یعنی در مثال بالا explore کردن v6 به علت بیشتر بودن v5 از max(min1,min2) نیاز نیست.

بخش c – هرس v11 و v12



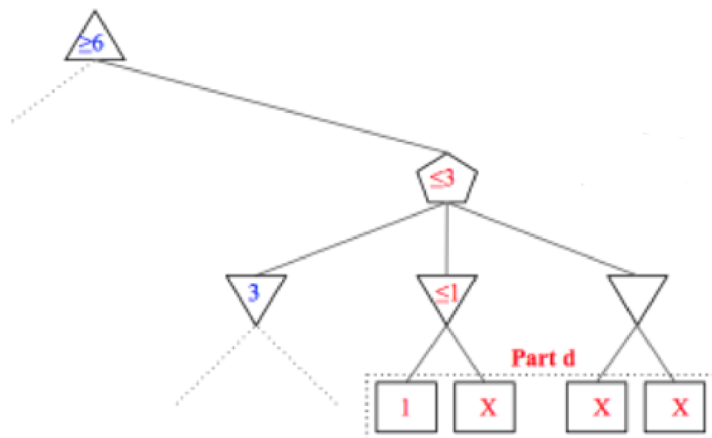
فرض کنید بالاترین ماکسیمم مقدار max1>= z را دارد. که از زیر درخت سمت چپش به دست آمده. سه مینیمم گیرنده در این بخش از درخت نیز min1,min2 و min3 هستند.

اگر z => max(v9,v10) => min3 باشد میتوانیم v11 و v12 را هرس کنیم. علت اینست که اگر هردو یا یکی از min2 و min3 نیز کمتر از z باشند، میانه گیر عددی کمتر از z میدهد و گفتیم بالاترین ماکسیمم مقدار بیشتر یا مساوی z دارد پس مقدار میانه روی آن تاثیری نخواهد داشت.

اگر min2 و min3 هر دو بیشتر از z باشند پس قطعا میانه بین آنهاست (چون min3 کمتر از z است) بنابراین نیاز نیست مقدار دقیق min3 را بدانیم پس نیاز نیست v11 و v12 را بررسی کنیم.

نکته قابل توجه اینست اگر v9 بزرگتر مساوی z باشد باید v10 حتما بررسی شود چون ممکن است v10 بزرگتر از v9 باشد و روی بالاترین ماکسیمم تاثیر بگذارد.

بخش d – هرس v14,v15,v16




فرض کنید در بخش c فهمیدیم min1 کمتر یا مساوی z است و در این بخش نیز مانند مثال درون شکل ضرفا با بررسی v13 میفهمیم min2 نیز کمتر از z است. اگر min3 بزرگتر از z باشد قطعا میانه بین min1 یا min2 که کمتر از z اند است پس میانه نیز کمتر از z است پس تاثیری روی بالاترین ماکسیمم ندارد. اگر min3 کمتر از z باشد پس میانه که بین min1,min2 و min3 است نیز کمتر از z است پس باز هم تاثیری روی بالاترین ماکسیمم ندارد. پس نیازی به بررسی ادامه گره ها نیست.