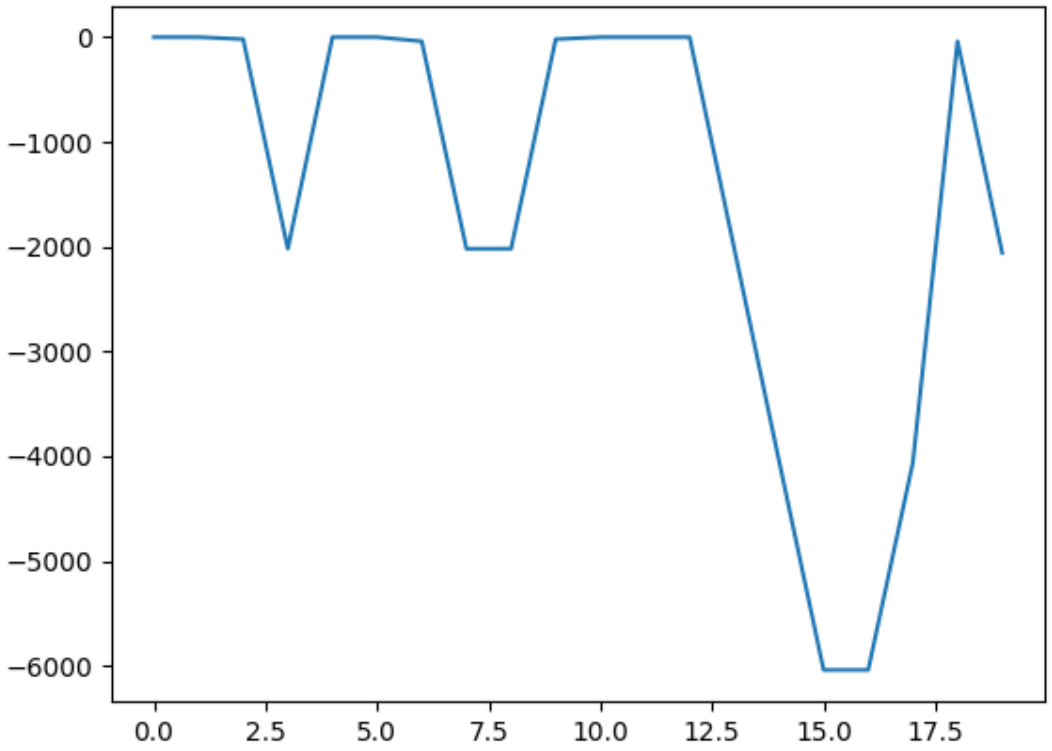
مهدی سلمانی صالح‌آبادی ۹۸۱۰۵۸۲۴

حالت اول) نفر اول رندوم، نفر دوم رندوم (نفر دوم برد)

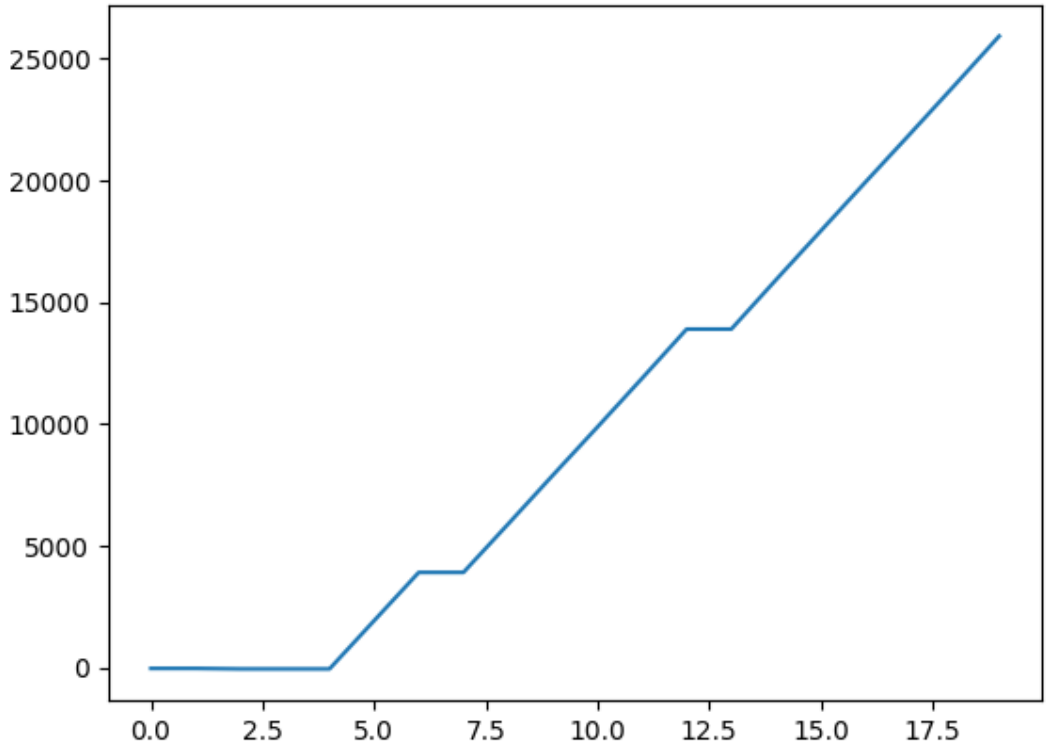
Result: -1 Client1_Delay: 0.0 Client2_Delay: 0.001



چون هر دو نفر رندم بازی کردند زمانی برای تصمیم گیری استفاده نکردند. (دیلی جفتشون تقریبا صفر است) و نتیجه هم تصادفی است یعنی اگر یک بار دیگر بازی انجام می‌شد ممکن بود نفر اول ببرد.

حالت دوم) نفر اول رندوم، نفر دوم minmax با عمق یک (نفر اول برد)

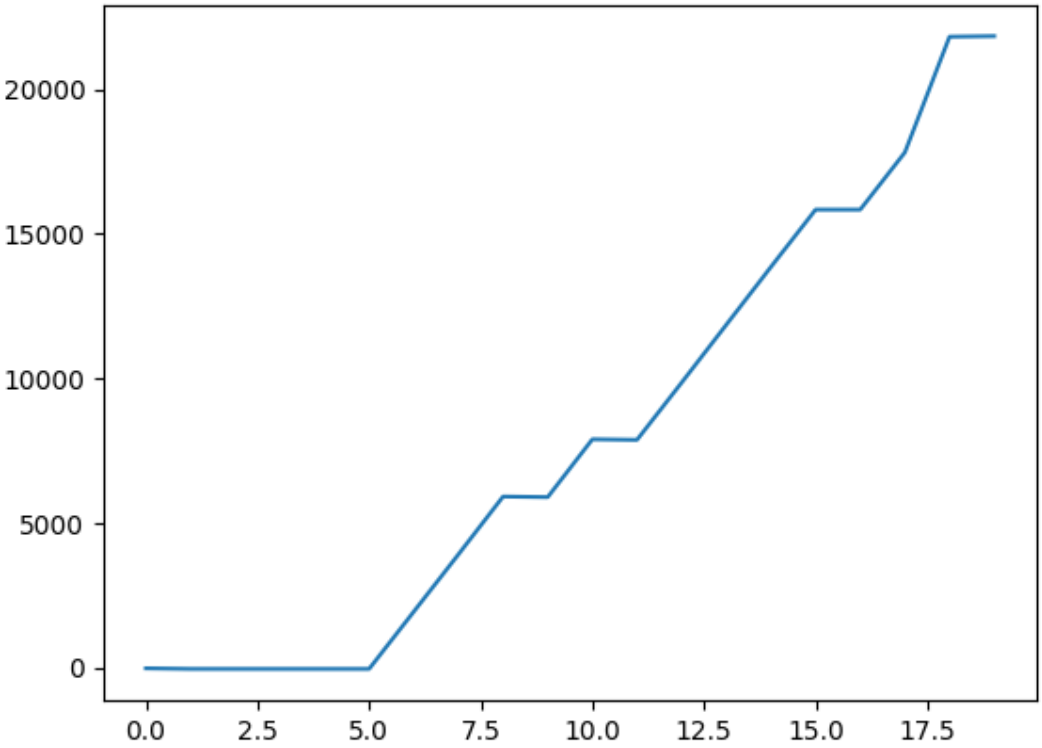
Result: 1 Client1_Delay: 0.0 Client2_Delay: 0.005



اینکه نفر دوم به دلیل بررسی بهترین انتخاب خود زمان صرف کرده واضح است. ولی چرا اینقدر بد شکست خورد؟ چون عمق بررسی این فرد کم است، نمی‌تواند حالاتی را که در آینده برای او در دو حرکت ۱۰۰۰ امتیاز میاورند را در ابتدا تشخیص دهد و برای همین دید محدود سعی می‌کند بیشتر ۱۰ امتیاز ۱۰ امتیاز، به امتیاز خود بیفزاید. (اگر جدول بازی را نگاه کنید کل مهره‌ها را در یک ستون قرار می‌دهد زیرا با بینش عمق یکی گذاشتن مهره در ستون بقلی (که بعدا می‌تواند منجر به ۱۰۰۰ امتیاز شود) توجیه ندارد.)

حالت سوم) نفر اول رندوم، نفر دوم pruning با عمق یک (نفر اول می‌برد)

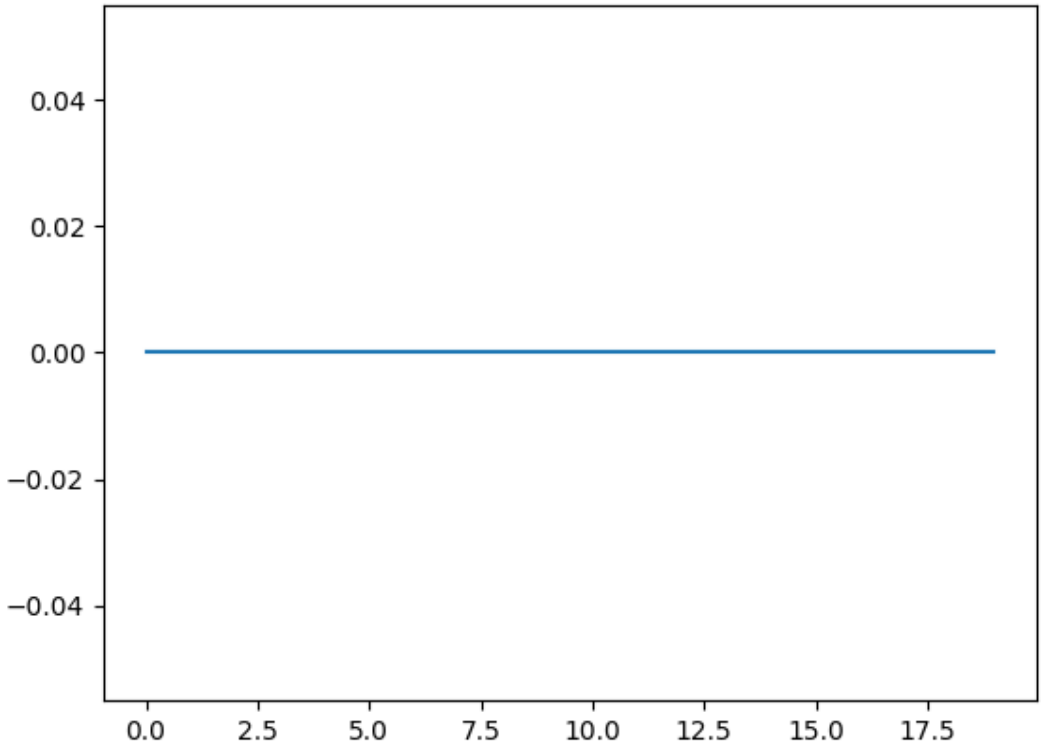
Result: 1 Client1_Delay: 0.001 Client2_Delay: 0.006



دلیل برد نفر اول در قسمت قبل بیان شد. از نظر سرعت اما انتظار pruning سریعتر باشد. (از minmaxاصلی) ولی اینگون نشد. چرا؟ چون در عمق یک اصلا تاثیری ندارد و باعث کهش تعداد حالات بررسی نمی‌شود.

حالت چهارم) نفر اول orginal_minmax و نفر دوم pruning هر دو با عمق یک (بازی مساوی شد)

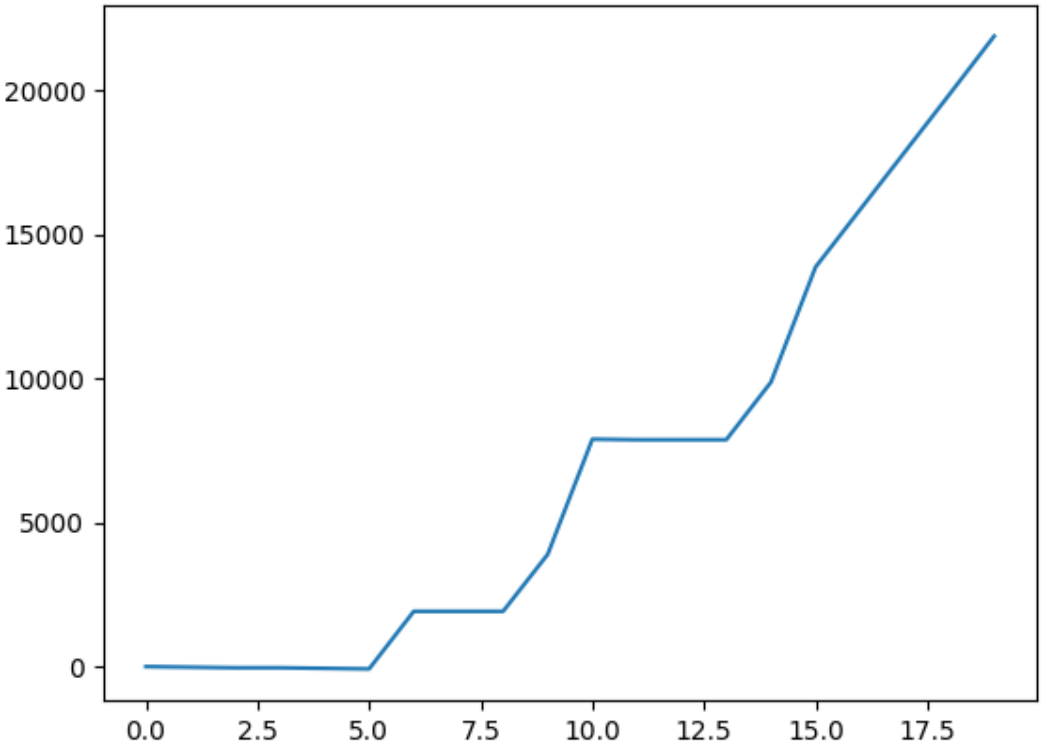
Result: 0 Client1_Delay: 0.005 Client2_Delay: 0.005



اینکه از نظر زمانی چرا بهبود نداریم در قسمت سوم بیان شد. اما چرا بازی مساوی شد. چون عمق جواب کم است هر دو نفر تمام مهره‌هایشان را در یک ستون قرار دادند و نتیجه اینگونه شد. (در واقع متقارن بازی کردند.)

حالت پنجم) نفر اول رندوم، نفر دوم minmax با عمق دو (نفر اول برد)

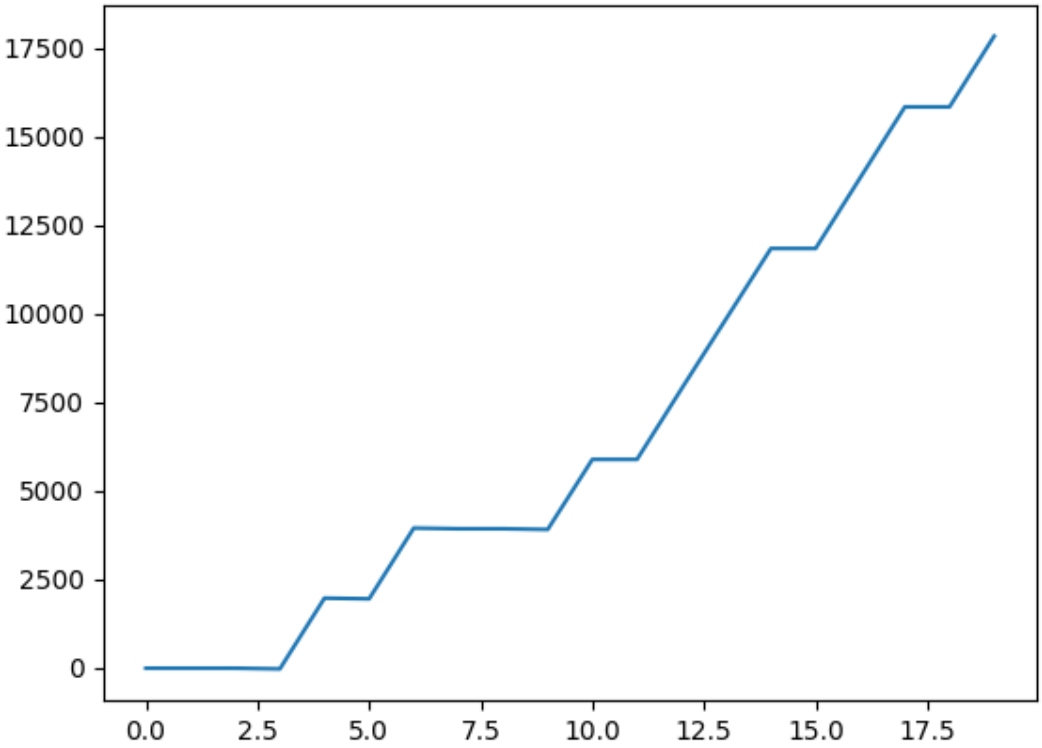
Result: 1 Client1_Delay: 0.001 Client2_Delay: 0.046



دقیقا مانند حالت دوم است. البته به دلیل عمق بیشتر از حالت دوم زمان تاخیر نفر دوم بیشتر شده‌است ولی دوباره چون نمی‌تواند با عمق دو امتیاز ۱۰۰۰ را تشخیص دهد (از ابتدا) تمام مهره‌ها را روی یک ستون قرار می‌دهد. (در واقع عمق دوم فقط باعث می‌شود که حرکت حریف را حدس بزند و باعث نمی‌شود که خودش بتواند به گونه‌ای مهره بگذارد که امتیاز ۱۰۰۰ را بگیرد.)

حالت ششم) نفر اول رندوم، نفر دوم pruning با عمق دو (نفر اول می‌برد)

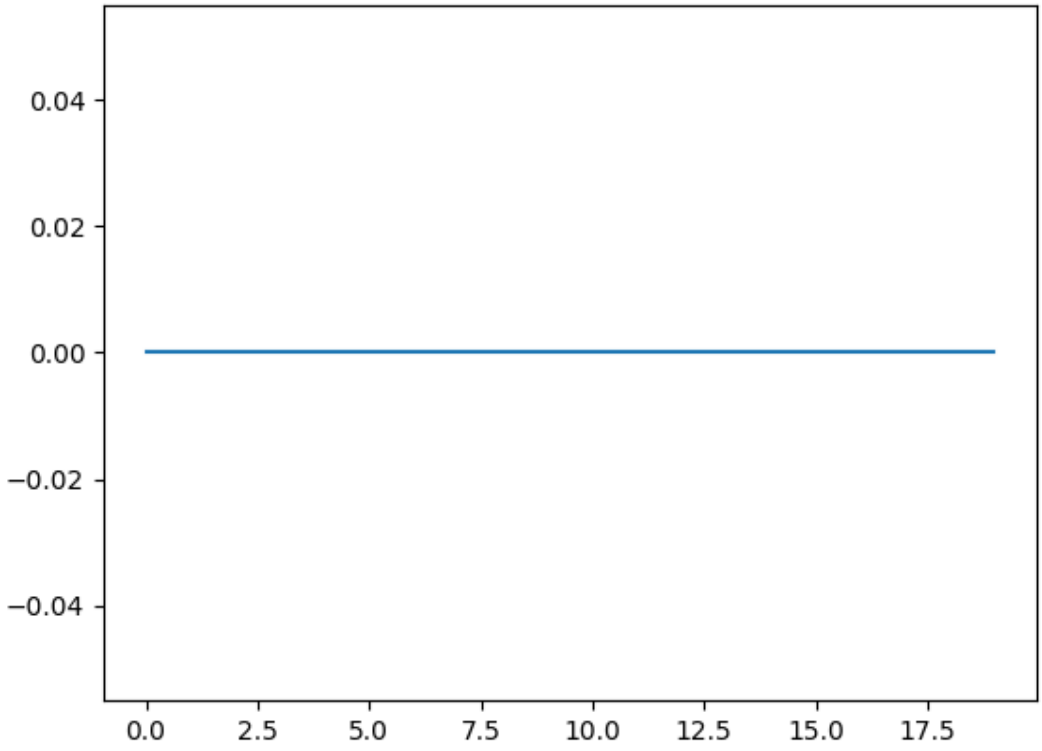
Result: 1 Client1_Delay: 0.001 Client2_Delay: 0.08



دلیل اینکه نفر اول می‌برد همان دلیل حالت قبلی است. اما چرا از نظر زمانی دوباره بهبود نداریم؟ چون دوباره روش pruning تا عمق دو روش کارایی نیست و بیشتر باعث اتلاف وقت می‌شود.

حالت هفتم) نفر اول orginal_minmax و نفر دوم pruning هر دو با عمق دو (بازی مساوی شد)

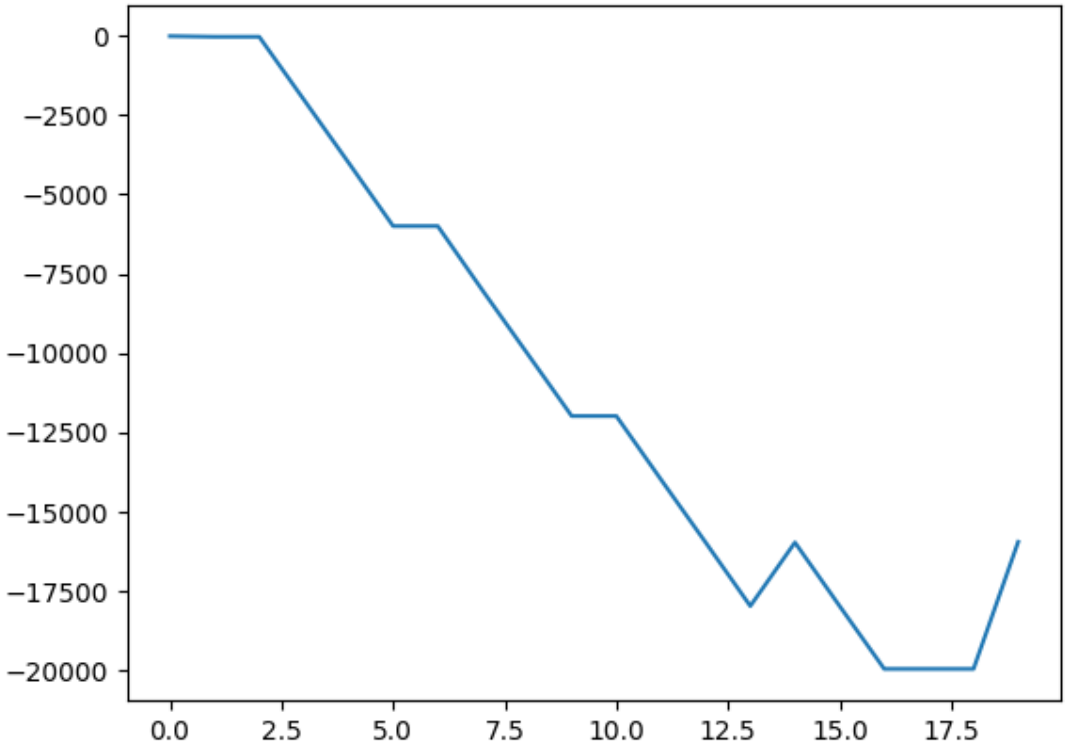
Result: 0 Client1_Delay: 0.172 Client2_Delay: 0.174



دقیقا مانند حالت چهارم است. هم دلیل مساوی‌شدن و هم دلیل بهبود نیافتن زمانی الگوریتم pruning

حالت هشتم) نفر اول رندوم، نفر دوم minmax با عمق سه (نفر دوم برد)

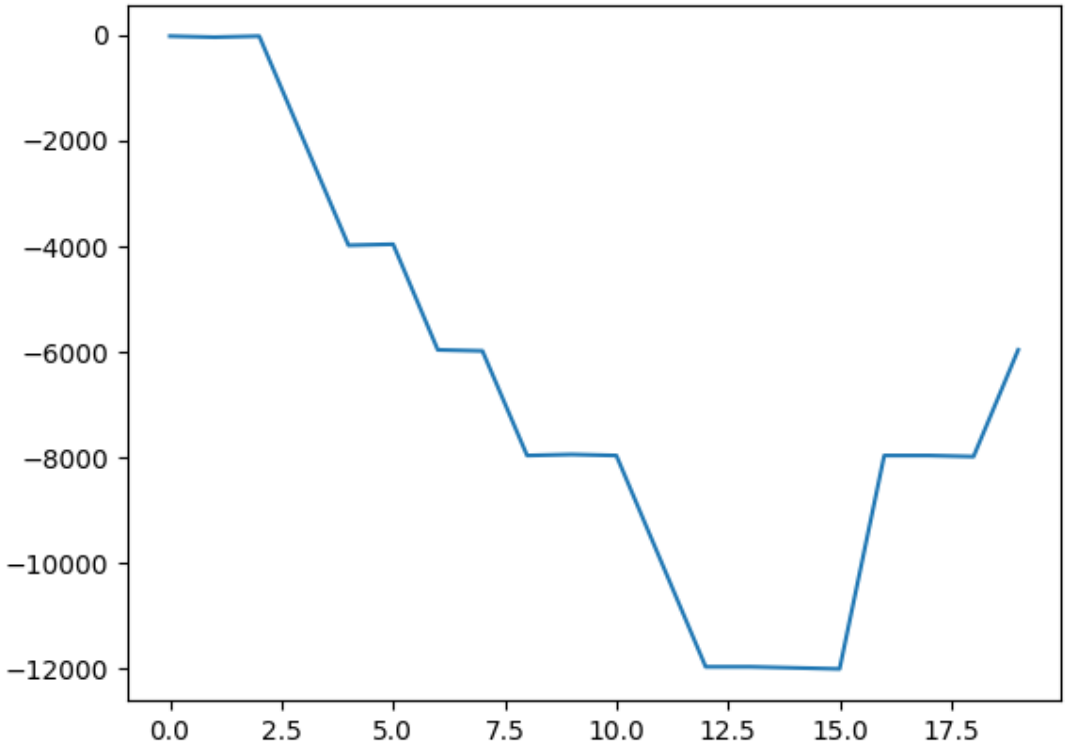
Result: -1 Client1_Delay: 0.001 Client2_Delay: 1.056



از این‌جا به بعد نفر دوم توانایی روش گرفتن ۱۰۰۰ امتیاز را دارد و دیگر در یک ستون مهره نمی‌گذارد.
برای همین با اختلاف نفر اول که رندوم بازی می‌کرد را شکست می‌دهد.

حالت نهم) نفر اول رندوم، نفر دوم pruning با عمق سه (نفر دوم می‌برد)

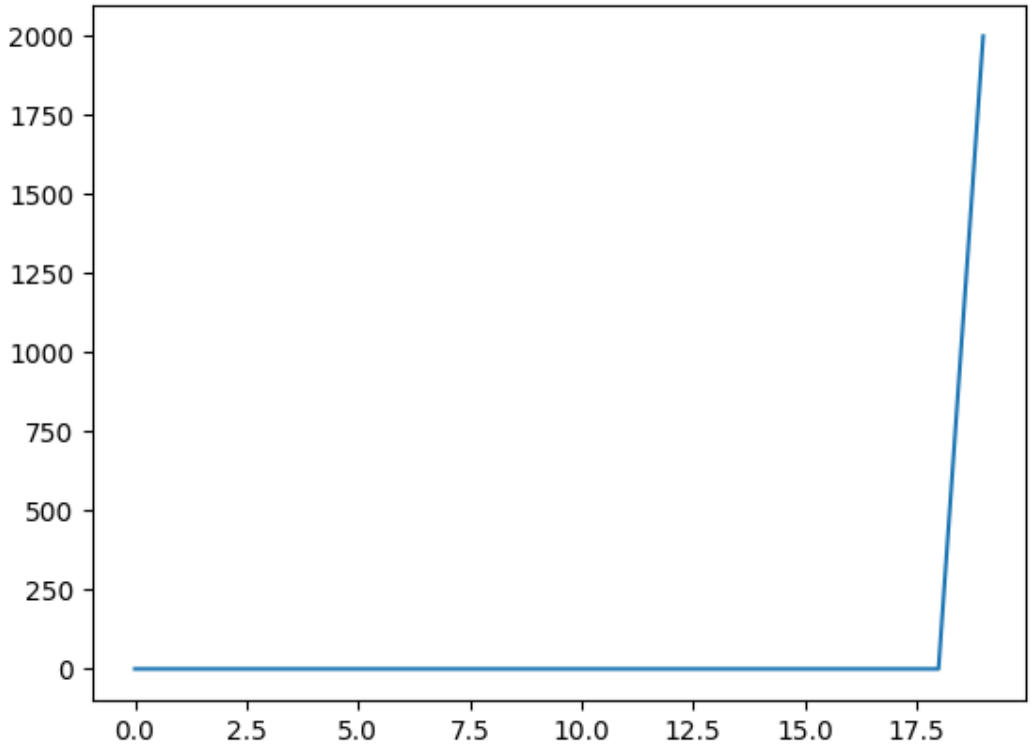
Result: -1 Client1_Delay: 0.0 Client2_Delay: 0.414



دلیل برد نفر دوم مانند حالت قبلی است. اما دقت کنید بالخره ما تاثیر pruning را در الگوریتم مشاهده کردیم که باعث افزایش سرعت minmax تا عمق ۳ شد و از نظر زمانی نصف original minmax زمان برد.

حالت دهم) نفر اول orginal_minmax و نفر دوم pruning هر دو با عمق سه (نفر اول برد)

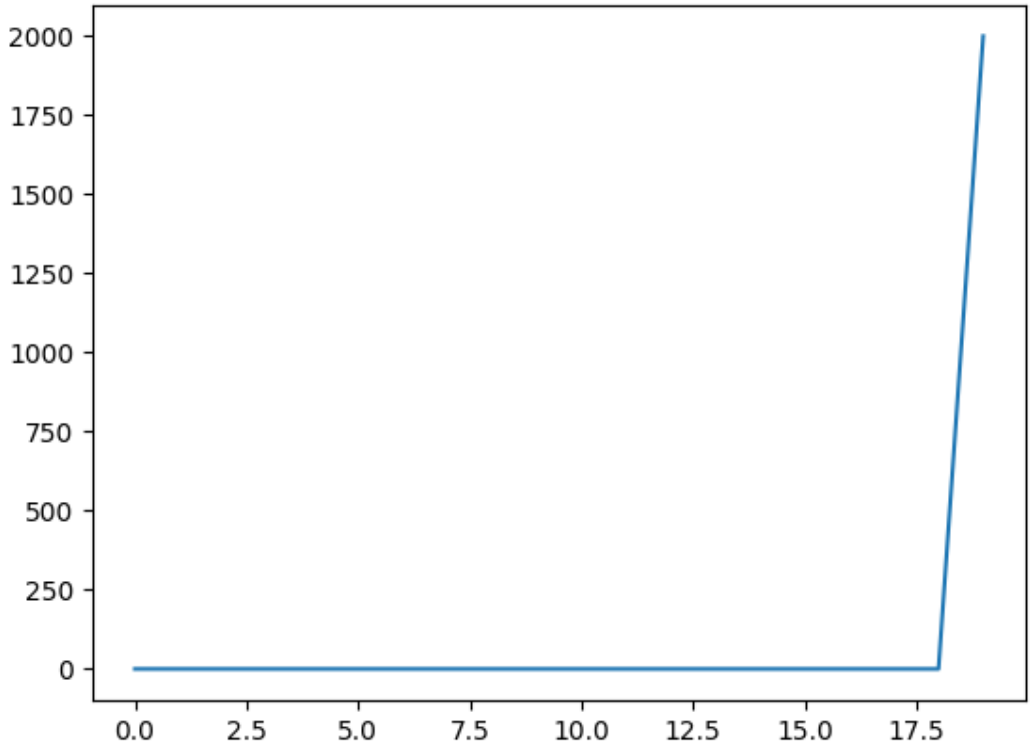
Result: 1 Client1_Delay: 1.811 Client2_Delay: 1.493



از نظر زمانی همانطور که انتظار می‌رفت نفر دوم سریعتر است. اما چرا نفر اول برد و بازی مساوی نشد؟ چونکه دو نفر در این حالت مهره‌هایشان را در یک ستون نمی‌گذارند و برای همین درست است که متقارن بازی می‌کنند ولی جایی می‌رسد که هر دو یک خانه را می‌خواهند و خب در این حالت نفر اول که بازی را شروع کرد دست پیش را دارد.

حالت یازدهم) نفر اول pruning با عمق چهار و نفر دوم pruning با عمق سه (نفر اول برد)

Result: 1 Client1_Delay: 23.199 Client2_Delay: 1.622



در این حالت نیز مانند حالت قبلی تا جایی از باز دو طرف متقارن بازی می‌کنند ولی خانه‌ای در صفحه به وجود می‌آید که جفتشان آن خانه را می‌خواهند و ابتکار دست نفر اول است. (دقت کنید افزایش عمق کمی تاثیرگذار است ولی دلیل اصلی برد نفر اول، نفر اول بودن است)

از نظر زمانی هم دقت کنید که اگر minmax با عمق ۴ انجام می‌دادیم میزان تاخیر خیلی خیلی زیاد می‌شد در حدی که احتمالا برنامه را terminate می‌کردیم.