**שאלה 1**

השחקן SimplePlayer בוחר את הכיוון בו יהיו לו את המספר המינימאלי של צעדים אפשריים לבחור, אך לא אפס, לפי הסדר: למעלה,ימינה, למטה ושמאלה.

**שאלה 2**

עבור הלוח הנ"ל, כאשר האדום הוא שחקן SimplePlayer והכחול שחקן כלשהו:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

השחקן SimplePlayer ייבחר ללכת למעלה כי יש רק צעד אחד אפשרי מהמשבצת הזו וזו האופציה הכי טובה מבחינת הגדרת ה SimplePlayer מכך שהוא בוחר ללכת למעלה קודם, וייקח את הפרי.

לאחר מכן יילך שמאלה וייקח את הפרי כי זה הכיוון היחיד שהשחקן שיכול ללכת.

לאחר מכן יילך למטה וייקח את הפרי כי זה הכיוון היחיד שהשחקן שיכול ללכת.

השחקן לקח את כל הפירות, והשחקן היריב נתקע בלי אפשרות ללכת ולכן ישלם את ה Penalty score, לכן שחקן ה SimplePlayer השיג את הפרש הנקודות המירבי ולכן פעל בצורה אופטימאלית כנדרש.

**שאלה 3**

היתרונות של ההיורסטיקה הזאת, היא שהיא מביאה לרווח מיידי הכי מהר.

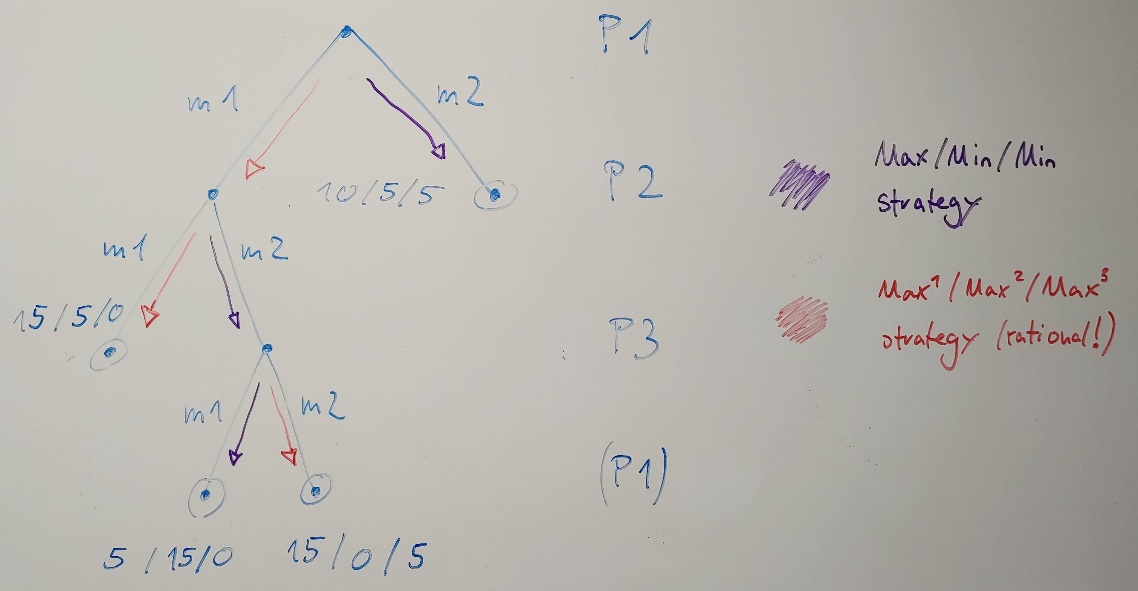
החסרונות של ההיורסטיקה הזאת, היא שאם יש כמה פירות עם אותו מרחק מנהטן מינימאלי היא לא בודקת את הרווח של הפרי, לכן גם הרווח המיידי אינו מקסימאלי.   
חסרון נוסף, ההיורסטיקה אינה מתחשבת במקרים שלאחר לקיחת הפרי נגיע למצב ללא מוצא, ואינה מסתכלת על שאר הפירות בלוח, לכן היא תוכל להוביל למצבים עם ערך נמוך בהרבה מהערך האופטימלי.

**שאלה 4**

??

**שאלה 5**

1. חסרון אחד ומרכזי, הוא ההנחה שהשחקנים לוקחים את הצעדים שיביאו למינימזציה את הרווח של השחקן שמריץ את המינימקס, ולא חושבים על הרווח האישי שלהם. דוגמא טובה לכך (הלקוחה מ Stack Overflowועבור המסלול הסגול):



במקרה כזה האלגוריתם יניח כי P2 ייקח צעד ימינה בעץ, על מנת למזער את הרווח של P1, ואז עבור P3 האלגוריתם יניח כי ייקח צעד (לא יעיל) שיביא לו רווח של 0 במשחק כשכל לקחת צעד ימינה בעץ על מנת לקבל רווח של 5.

חסרון נוסף, הוא אי התחשבות בשיתוף פעולה בין שחקנים על מנת לבצע טקטיקות רלוונטיות למשחק, כמו מזעור השחקן הכי טוב וכו'

1. אסטרטגיה חלופית שנציע, היא שבכל שלב במינימקס, במקום למזער את תועלת השחקן הראשון, כל שחקן ימקסם את התועלת שלו.

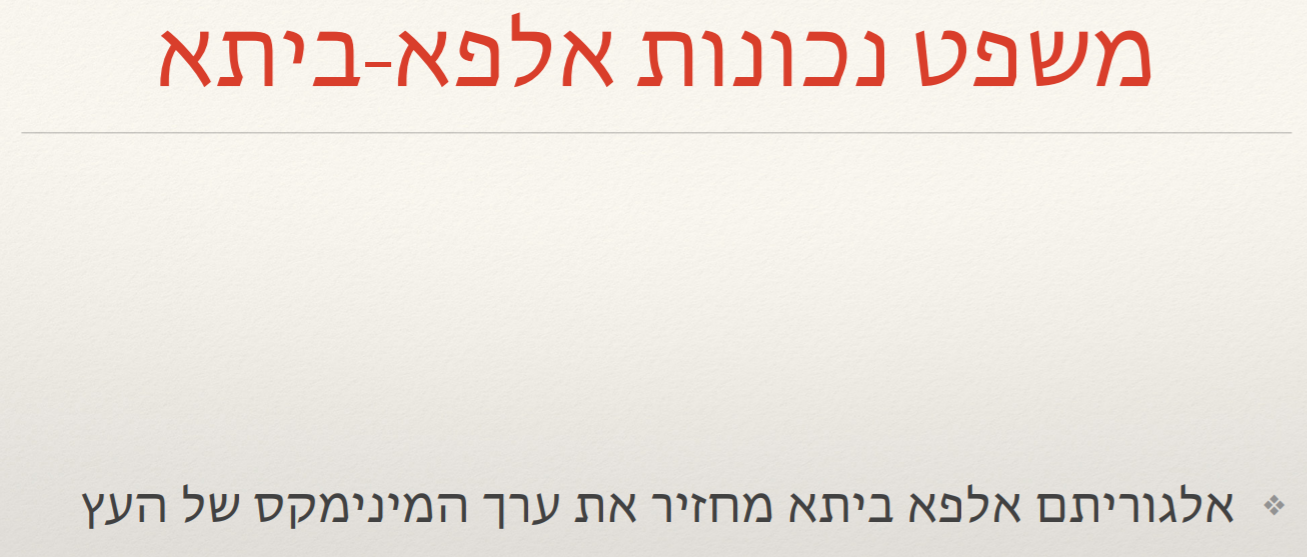
במקרה כזה, השחקנים יבחרו צעדים יותר "הגיוניים" והם ימקסמו את התועלת של עצמם.

במקרה כמו למעלה במסלול האדום, P3, יבחר צעד ימני בחלק העמוק בעץ, ואז P2 יבחר את הצעד השמאלי, ו P1 יבחר את הצעד השמאלי, ואז קיבלנו תוצא הגיונית, ש P3 בוחר צעד שיביא לו רווח עצמי, במקום למזער את רווח P1.

**שאלה 6**

1. כפי שראינו בתרגולים ובהרצאה, לא בהכרח alpha-beta יהיה יעיל מבחינת זמן ריצה, לדוגמא במקרה בו לא יתבצע כלל גיזום ונצטרך לפתח רגיל את תת העץ.

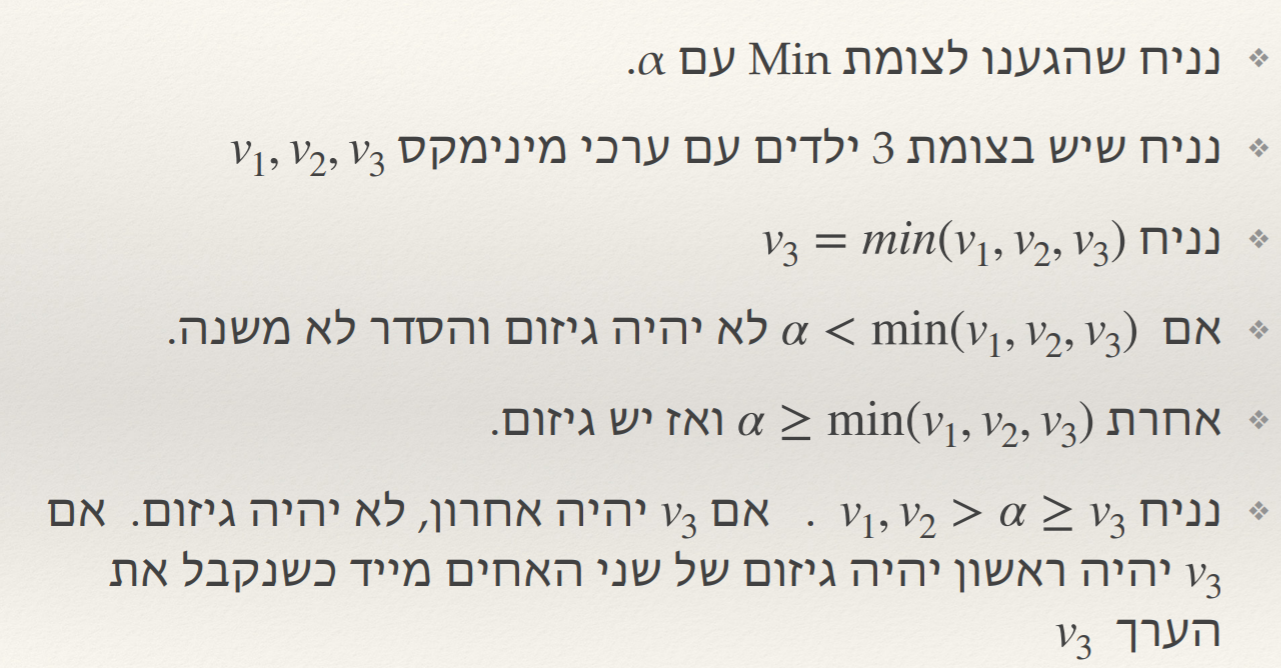
אך כפי שראינו בהרצאה, יש מקרים בהם יש שיפור בזמן הריצה:  


1. כפי שראינו בהרצאה (ללא הוכחה):

לכן ערך המינימקס עבור זהה למינימקס רגיל.

**שאלה 7**

1. לפי הדוגמא שראינו מההרצאה עבור סידור ילדים :



במקרה כזה, סידור הילדים יגרום לגיזום העץ, שזה אומר שנפתח פחות ילדים בתת העץ ולכן נחסוך בזמן הריצה.

1. האלגוריתם לא בהכרח יחזיר את אותן סדרת מהלכים.  
   אם קיימים לדוגמא שני צמתים עם אותו ערך היוריסטי ואותה תועלה, סידור הילדים לפי ההיורסטיקה לא יהיה חד-חד ערכי ונוכל לסדר את הילדים בשני סדרים שונים, ולפי הסדר שסודרו הבנים בתת העץ, נבחר את הראשון מביניהם, למרות שעבור סידור אחר, נבחר את השני.  
   לכן בחירת המהלך תהיה תלויה בסדר שהופיעו ב ללא הסידור לעומת העם הסידור.

**שאלה 8**

וריאציית ה Anytime Contract של Minimax, מציינת את משפחת האלגוריתמים שמקבלים זמן כקלט, כלומר להחזיר את ערך המינימקס הכי טוב שניתן תחת הגבלת זמן.

העמקה הדרגתית בהקשר זה, מגדירה כי נממש Resource bounded Minimax, כלומר מינימקס מוגבל עומק, ונריץ כל פעם עם הגבלת עומק גדולה יותר, עד שבעומק מסויים נגיע למגבלת הזמן במהלך חישוב האלגוריתם ונחזיר את הערך שקיבלנו עבור העומק הקודם.

**שאלה 9**

הבעיה בהעמקה הדרגתית שמוצגת בהרצאה, היא שנשקיע משאבים רבים בחיפוש עבור העומק שבו נגיע למגבלת הזמן, כלומר נפתח חלק רב מתת העץ אבל אז נעצור את החיפוש כשייגמר לנו הזמן.  
הפתרון המוצע בהרצאה הוא, שנשמור את ערך המינימקס עבור הבנים שפיתחנו בהרצה עם העומק שבו קטענו את האלגוריתם, ואז בריצות הבאות של המינימקס נחסוך את הפיתוח מחדש של הבנים האלו.

**שאלה 10**

נציע אלגוריתם מהצורה : כאשר :

עבור התור הזמן שיוקצה יהיה .

כאשר נמצא את לפני הרצת המינימקס בתור הראשון בלבד, באמצעות חיפוש Flood Fill (דומה ל BFS)מוגבל משאבים, משני השחקנים, ונספור כמה משבצות ישיגות.

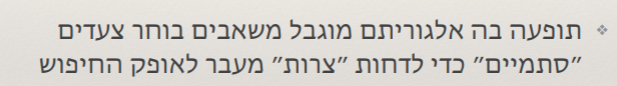
*הוא כלומר הזמן שנשאר לתורות.*

נחשב את X באמצעות סכום סדרה הנדסית:

בסיום תור נעדכן את X, באמצעות עדכון num\_turns ו remaining\_time

ההיגיון מאחורי האלגוריתם הוא שלקראת סוף המשחק, כלומר בתורות האחרונים לעומת הראשונים, עומק העץ נמוך בהרבה, לכן נרצה להקצות לתורות האחרונים פחות משאבים לעומת התורות הראשונים, בהם העץ גדול בצורה אקספוננציאלית.

**שאלה 11**

אפקט האופק מההרצאה:  


כלומר, בוחרים צעדים לא אופטימליים, מכיוון שהחיפוש מינימקס שלנו הוגבל בעומק בשביל לראות את הטעות בצעד שלו.

הפתרון המוצע בהרצאה הוא העמקה סלקטיבית, כלומר עבור צעדים משמעותיים במשחק, שניתן להגדירם למשל כמו שינוי גדול על ערך ההיורסטיקה (לדוגמא בשחמט, כל עוד יש לקיחת כלים), נפתח את עומק העץ עבור עלים שמייצגים צעדים אלו עד שנגיע להתייצבות ערך ההיורסטיקה.

עבור הסוכן שלנו, שילוב של פתרון זה יכול לתרום במקרים בהם ניקח פירות, שזה ייצג צעד משמעותי, או חסימת היריב לדוגמא.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

עבור פרי = 100 נקודות, עונש אי יכולת תזוזה = 1000 נק

במקרה כזה, עבור עומק חיפוש מוגבל ל2, השחקן האדום יילך לכיוון למטה בשביל להגיע לפרי, ואז ייתקע לפני השחקן הכחול, מספר הנקודות שלו יהיו -900 ויפסיד את המשחק, בעת, שאם היה הולך למעלה, היה מצליח להשיג תיקו עם השחקן היריב.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

במצב כזה עם עומק 2, עם הגדרות משחק זהות ללוח הקודם, השחקן האדום לא "רואה" בעומק העץ, שאם יילך למעלה יוכל לחסום את השחקן הכחול, ולנצח בוודאות את המשחק עם חסימה, ויילך לכיוון הפרי במקום, מה שלא יבטיח לו ניצחון.