**חלק א**

**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**

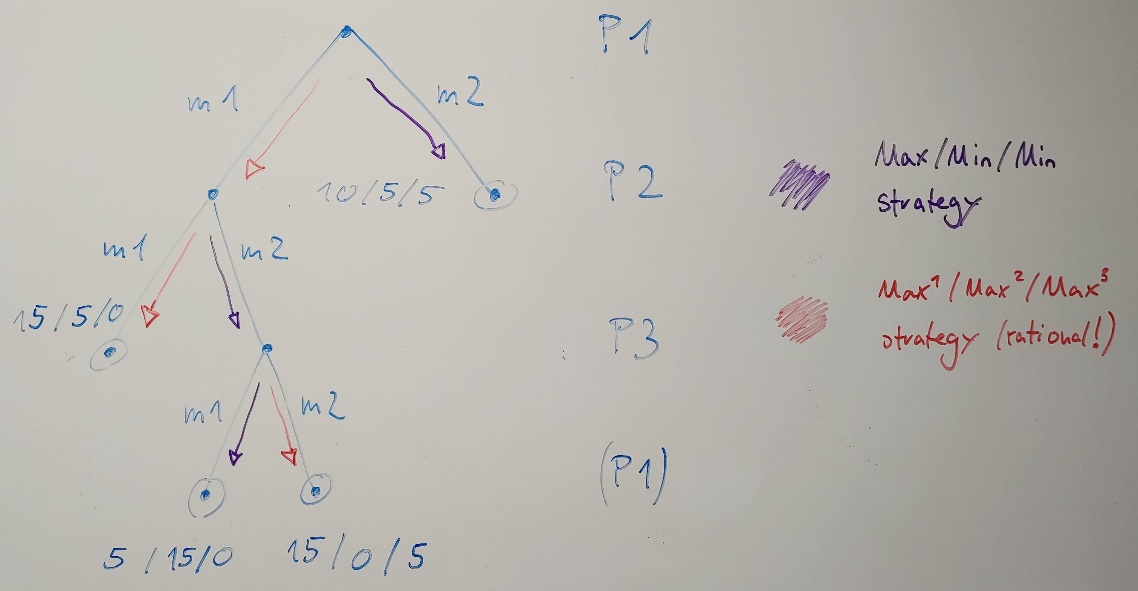
היוריסטיקה שלנו מורכבת מ4 מרכיבים:

* 1. היוריסטיקה הזאת פשוט מחשבת את הפרש הניקוד בין השחקן שלנו לשחקן היריב. כיוון שניצחון מוגדר לפי השחקן המסיים את המשחק עם הניקוד הגבוה, סביר להניח שהפרש ניקוד גדול וחיובי יוביל לסיכוי גבוה לניצחון.
  2. היוריסטיקה הזאת מחשבת את סכום יחסי שוויי הפירות שאנו מסוגלים להשיג למרחק שלנו מהם. השגת פירות היא דרך עיקרית להשגת נקודות ולכן רצוי להגיע למצבים מהם נוכל להשיג פירות יקרות, אך עם זאת אנו נעריך פירות רחוקים פחות מאשר קרובים כיוון שהן דורשות יותר טרחה/זמן להגיע אליהם (ולכן יכולות לבוא על חשבון אסיפת פירות אחרים). לכן צפוי שיוריסטיקה זו תעזור לשחקן שלנו לשערך את המצב שלו ביחס לניקוד הפירות שסביר להשיג מנקודה זו.
  3. יוריסטיקה זו מחשבת את ההופכי של המרחק שלנו לפרי הקרוב ביותר. יוריסטיקה זו מטרתה לגרום לשחקן שלנו לנסות להגיע למצבים בהם הוא יהיה קרוב לפרי כדי שיוכל להשיג אותן וכדי לחטוף אותן לפני שהיריב ישיג אותן. לכן יוריסטיקה זו מצופה גם להעלות את הניקוד שלנו וגם להוריד את ניקוד היריב, מה שיוביל לניצחון יותר מובטח.
  4. יוריסטיקה זו מחשבת את מספר המשבצות מסביב לשחקן שלנו שבו יוכל לצעוד בצעד הבא ומחסרת מזה את אותו הדבר על היריב. יוריסטיקה זו נועדה לעזור לשחקן שלנו להתמודד עם סוף המשחק בו חשוב לחסום את תנועת היריב ולהישאר במצב מרווח בעצמנו כדי להגיע למצב בו היריב יתקע וייאלץ לחטוף את הפסד הנקודות על היתקעות ראשונה. הפסד זה (והימנעות מהפסד זה בעצמנו) יגדיל את הפרש הנקודות בין השחקן שלנו לשחקן היריב על מנת להעלות את הסיכוי לניצחון. היוריסטיקה תעזור לו גם במהלך המשחק בכך שהיא תדאג שתמיד יהיו לו הרבה כיווני תנועה פתוחים לעומת היריב בתקווה שחלק מאלה יהיו מסלולים למצבים טובים.

אנו מחשבים סכום **ממושקל** של יוריסטיקות אלה כדי להגיע ליוריסטיקה שנותנת דגש כמה שיותר מתאים על כל רכיב, והתוצאה היא היוריסטיקה הסופית.

**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.

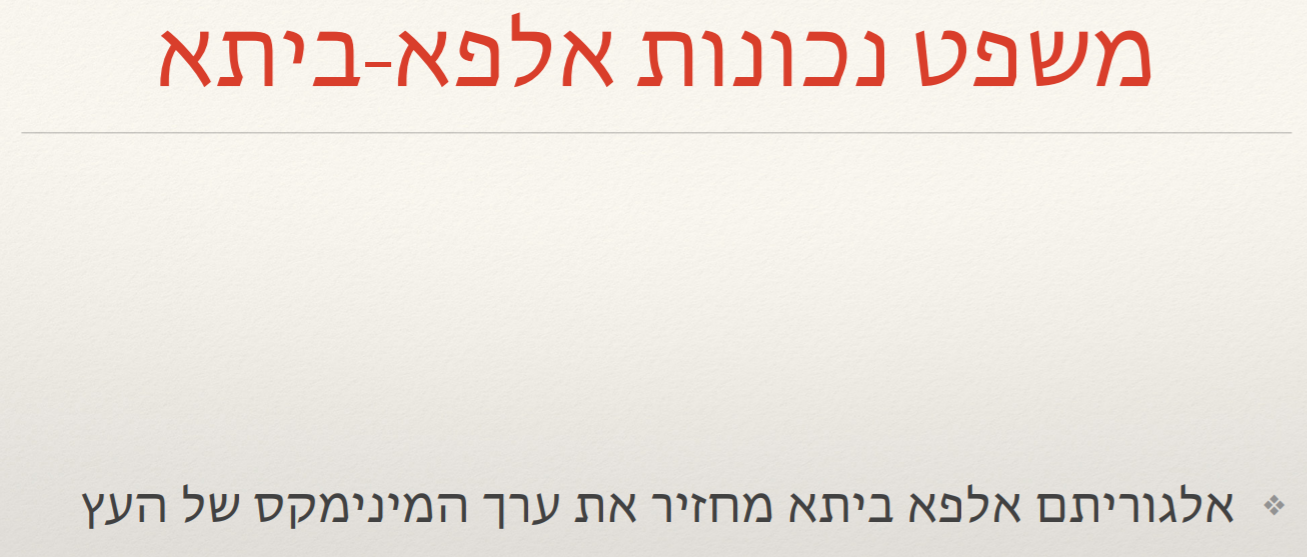
במקרה כזה, השחקנים יבחרו צעדים יותר "הגיוניים" והם ימקסמו את התועלת של עצמם.

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.

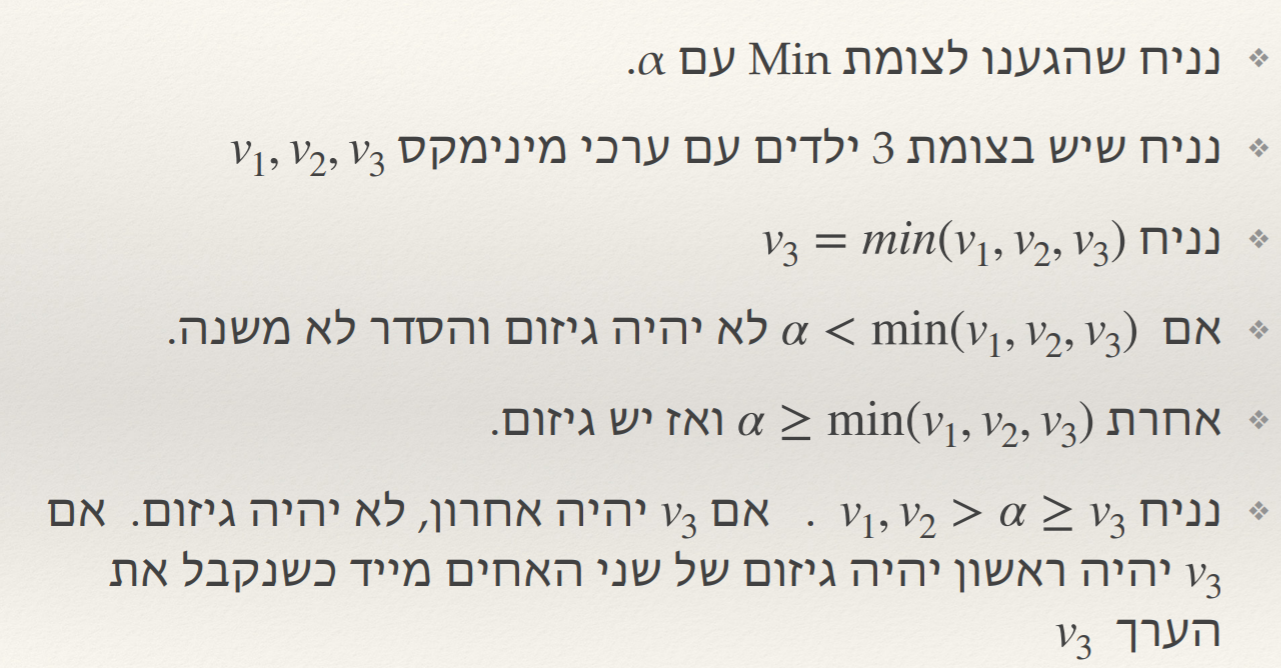
אך כפי שראינו בהרצאה, יש מקרים בהם יש שיפור בזמן הריצה:  


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:

כאשר נמצא את לפני הרצת המינימקס בכל תור, באמצעות חיפוש Flood Fill (דומה ל BFS), מהשחקן שלנו, ונספור כמה משבצות ישיגות.

*הוא כלומר הזמן שנשאר לתורות.*

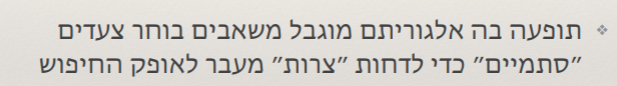
נחשב את זמן תור באמצעות :

בסיום תור נעדכן את .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.

באלגוריתם זה מתחשבים במשבצות הישיגות, כי אם יש לוח שרוב המשבצות בו לא ישיגות, האלגוריתם הנאיבי למשל לא מנצל בצורה אופטימאלית את הזמן.

**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.

**שאלה 12**

המתכנת יכול להריץ את אלג' אלפא-בטא עם כל הגיזומים הרגילים, ובנוסף עם התנאים הנוספים:

1. אלפא לא ירד מערך המינימקס. כלומר נחליף את עדכון האלפא באלגוריתם המוצג בהרצאה מ:  
      
   ל:  
   .
2. בטא לא יעלה על ערך המינימקס. כלומר נחליף את עדכון הבטא באלגוריתם המוצג בהרצאה מ:  
   beta   
   ל:  
   beta

האלגוריתם המתוקן יגזום יותר מאשר האלגוריתם המקורי על ידי כך שצמתי עץ החיפוש המייצגים את התורות של שחקן המקסימום ייגזמו ברגע שיעלו על ערך גדול מערך המינימקס כי מובטח שאלה לא ייבחרו למינימקס. וצמתי עץ החיפוש המייצגים את התורות של שחקן המינימום ייגזמו ברגע שיעלו על ערך קטן מערך המינימקס, מאותו סיבה. במקרה הכללי כבר 2 רמות לפני הרמה העמוקה ביותר כל הצמתים כבר יחזירו את ערך המינימקס (כיוון שערכים אלו נבחרים רק מבין ערכים שגדולים/קטנים (בהתאמה לסוג התור) מערך המינימקס והם רק יחזירו ערכים שקטנים/גדולים מערך המינימקס (בהתאמה) שזה מוציא את כל הערכים שאינם המינימקס. ומובטח מההנחה שקיים מסלול חיפוש שמחזיר את המינימקס). המסלול חישוב הראשון שמצליח להגיע לשורש העץ (המייצג את התור הנוכחי) יקבע את הצעד שיש לקחת על מנת להשיג את ערך המינימקס. במקרה הטוב ביותר ערך המינימקס מוחזר מהמסלול הראשון אותו מנסים, ובמקרה זה אלגוריתם מינימקס ירד דרך מסלול אחד ישר לערך המינימקס ויחזור ישר על המסלול הזה מבלי לפתח מסלולים חדשים נוספים. במקרה הגרוע ביותר ערך המינימקס מוחזר רק מהמסלול האחרון אותו מפתחים, ובמקרה זה רק המסלול האחרון יצליח לפעפע את ערכו עד לשורש העץ (וכל השאר יחזירו ערכים אינסופיים בלבד ולכן לא יבחרו).

**שאלה 13**

ניתן במקרה זה לבצע את כל הגיזומים הסטנדרטיים של אלגוריתם אלפא-בטא ובנוסף:

1. כל צומת בעץ החיפוש המייצג את שחקן היריב ("שחקן המינימום") יחזיר מיד את הערך -5 (יחד עם הצעד הנלווה לבן שהחזיר ערך זה) אם אחד הבנים שלו החזיר את הערך -5, וזאת במקום להמשיך לחשב את ערכי החזרה של שאר בניו לחפש ערך קטן עוד יותר מ- -5 (כי לפי ההנחה, לא קיימת ערך כזה ולכן מובטח שלא תהיה בן שיחזיר ערך קטן מזה, ולכן הצעד שהחזיר והערך -5 הן האופטימליים מבחינתו). בכך האלגוריתם בעצם גוזם את שאר הבנים שהיה עתיד לחשב.
2. כמו כן, כל צומת בעץ החיפוש המייצג את שחקן המינימקס ("שחקן המקסימום") יחזיר מיד את הערך 5 (יחד עם הצעד הנלווה לבן שהחזיר ערך זה) אם אחד הבנים שלו החזיר את הערך 5, וזאת במקום להמשיך לחשב את ערכי החזרה של שאר בניו לחפש ערך גדול עוד יותר מ- 5 (כי לפי ההנחה, לא קיימת ערך כזה ולכן מובטח שלא תהיה בן שיחזיר ערך גדול מזה, ולכן הצעד שהחזיר והערך 5 הן האופטימליים מבחינתו). בכך האלגוריתם בעצם גוזם את שאר הבנים שהיה עתיד לחשב.
3. כל צומת בעץ החיפוש המייצג תוצאה אקראית במהלך המשחק (צומת המייצג את "השחקן הסטוכסטי") שמגיע לאחר צומת של שחקן המקסימום יתחיל את חישוב התוחלת שלו תחת ההנחה שכל הפעולות שיקח יחזירו ערך 5 (ולכן התוחלת שווה 5) ויעדכן את ערך התוחלת שעתיד להחזיר בכל החזרה של ערך מאחד מבניו. כאשר התוחלת הרצה יורדת מאלפא (או שווה לה) יפסיק מיד את חישוב התוחלת שלו (יגזום את שאר בניו שעוד לא החזירו את ערכיהם) ויחזיר (כדי שלא ייבחר על ידי הצומת שמעליו), וזאת מכיוון שגם אם כל שאר בניו יחזירו את הערך הגבוה ביותר (5) כפי שהניח א פריורי (וקל וחומר שכל ערך קטן מזה), תוחלתו לא תעלה על הערך שלה ברגע הגיזום, ולכן כיוון שערך זה קטן/שווה לאלפא מובטח שהשחקן מקסימום שמעליו לא יבחר בה ולכן אין צורך להמשיך לחשב את ערכה כי אין לו השפעה.
4. כמו כן, כל צומת בעץ החיפוש המייצג תוצאה אקראית במהלך המשחק (צומת המייצג את "השחקן הסטוכסטי") שמגיע לאחר צומת של שחקן המינימום יתחיל את חישוב התוחלת שלו תחת ההנחה שכל הפעולות שיקח יחזירו ערך -5 (ולכן התוחלת שווה -5) ויעדכן את ערך התוחלת שעתיד להחזיר בכל החזרה של ערך מאחד מבניו. כאשר התוחלת הרצה עולה על בטא (או שווה לה) יפסיק מיד את חישוב התוחלת שלו (יגזום את שאר בניו שעוד לא החזירו את ערכיהם) ויחזיר (כדי שלא ייבחר על ידי הצומת שמעליו), וזאת מכיוון שגם אם כל שאר בניו יחזירו את הערך הקטן ביותר (-5) כפי שהניח א פריורי (וקל וחומר שכל ערך קטן מזה), תוחלתו לא תרד מהערך שלה ברגע הגיזום, ולכן כיוון שערך זה גדול/שווה לבטא מובטח שהשחקן מינימום שמעליו לא יבחר בה.
5. כל צומת בעץ החיפוש המייצג תוצאה אקראית במהלך המשחק המגיע לאחר תור נוסף של השחקן האקראי ייחשב למתחת לתור של המעל תור השחקן הסטוכסטי שמעליו (כי שחקן זה הוא זה שיבחר האם לקחת או לעזוב את תוצאת הבחירה הסטוכסטית).

**שאלה 14**

1. הרצת מינימקס לעומק D+1 לוקח את כמות הזמן שלוקח למינימקס לרוץ על כל B בניו של שורש עץ החיפוש לעומק D, סה"כ זמן ריצה לפי ההנחות של (B חיפושים שכל אחד לוקח זמן ) דקות. בכך ייגמר לו כל הזמן לחישובי מינימקס בשאר B-1 המהלכים הנותרים ולא יוכל להשתמש במינימקס כדי לחשב בהם צעדים מושכלים.
2. 1. עליו לשמור קשת אחת לכל תור שלו (הצעד הטוב ביותר שמחזיר את ערך המינימקס החל מנקודה זו) ו-B קשתות לכל תור של היריב שלו (על מנת לכסות את כל אפשרויות הצעדים של היריב). כיוון שהוא מתחיל את מינימקס על התור שלו והתורות באות לסירוגין, ישנם שכבות בעץ של התורים של שחקן המקסימום ו שכבות של תורות שחקן המינימום, סה"כ:  
      מספר קשתות של שחקן המינימום=  
      מספר קשתות של שחקן המקסימום (השחקן שאנו דנים בו)=
   2. כעת תורות יכולות להיות מושכלים ביותר בכך שהן מיודעות על ידי ביצוע אלגוריתם מינימקס עד לעומק של תחתית העץ שחושב (צומת בעומק l משורש עץ החישוב יתעדכן על ידי הרצת מינימקס לעומק D+l-1).
   3. לשחקן משופר זה יהיה קירוב טוב יותר לערך המינימקס האמיתי (של הרצת האלגוריתם עד תחתית עץ החיפוש ולא רק עד לעומק D) תחת ההנחה שההיוריסטיקה מונוטונית בתור הראשון, ולכן יוכל לבצע צעד יותר מושכל ממנו בתור זה. עם זאת, שמירת כל העץ הזה דורש הרבה מקום (סיבוכיות מקום אקספוננציאלי ב-D לעומת סיבוכיות המקום O(1) הדרוש למינימקס רגיל ולכן לא ניתן ליישום כאשר חסר מקום לכך). בנוסף, בתור ה-l השחקן המשופר יתעדכן לפי צעד המינימקס הנבחר לפי הרצת אלגוריתם מינימקס לעומק D+l-1 לעומת שחקן המינימקס הרגיל שתמיד בוחר בצעד מינימקס לפי הרצת מינימקס לעומק D, ולכן בתור השני השחקן המשופר והשחקן מינימקס הרגיל יבחרו צעד לפי עץ מינימקס בעל אותו עומק (ולכן ישחקו באותו רמת השכלה) ובתורות הבאים כבר שחקן המינימקס הרגיל יבחר בצעדים יותר מושכלים עקב התבססותו במינימקס המורץ לעומק גדול יותר מהמשופר. לכן, כל עוד התור הראשון אינו משמעותי יותר מאשר סכום המשמעויות (לפי פונקציית "משמעות" התלוייה בערך המינימקס של צומת) של שאר B-1 התורות, סביר להניח ששחקן המינימקס הרגיל ישחק באופן מושכל יותר. (ואחרת השחקן המשופר ישחק באופן מושכל יותר עקב בחירת הצעד המשמעותי ביותר באופן המושכל ביותר).

**חלק ו**

**שאלה 1**

שחקן המינימקס שלנו משתמש ביוריסטיקה שתוארה בחלק א שאלה 4 הלוקחת בשיקול מרכיבים שונים של ניסיון לקבל ניקוד גבוה מהיריב בכל דרך שהיא, מרכיב הגעה למצב ממנו ישנם אופציות טובות להשגת פירות, מרכיב חטיפת פירות מהירה מהאויב ומרכיב חסימת תנועת האויב ביחס לתנועה שלנו כדי לתת לנו אפשרויות להגעה ליותר מצבים (וביניהם הטובים).

**שאלה 2**

השחקן שמימשנו עבור התחרות הינה אותה שחקן מסעיף ד, המשתמש באותה שיטת חלוקת זמן המוסבר מטה בסעיף 3.

**שאלה 3**

ניהול הזמן עבור הגבלת זמן לתור הסתמכה בכך שהנחנו שזמן החישוב העיקרי מתבצע בחישובי היוריסטיקה וששבריר שנייה קבועה תספיק לחזרה מעומק החיפוש הארוך ביותר שסביר שהשחקן יגיע אליו (לא רצינו לבצע חישובי זמנים מסובכים כדי להימנע מדדלוק בו דרוש זמן כדי לחשב אסטרטגיות ניהול זמן). השתמשנו באלגוריתם interruptable AlphaBeta שהפסיק את החיפוש כאשר נותר לו רק כמות הזמן המוקצב לחזרה מהרקורסיה כדי להגיע לעומק רקורסיה אופטימלית שניתן להגיע אליה תוך הזמן המוקצב כדי לחשב אסטרטגיה. ניסויים אמפיריים רבים הראו לנו כי אכן (לפחות על המחשבים שלנו) זאת אסטרטגיה מנצחת.

על מנת להתמודד עם מגבלת זמן גלובלית החלטנו קודם לחלק בכל תור את שארית הזמן הגלובלי שווה בשווה בין כל תורות המשחק הנותרים (חלוקה פשוטה של ) כאשר ההערכה שלנו למספר התורות הנותרות מכל מצב נקבעה על ידי ביצוע חיפוש BFS לאורך הלוח מהשחקן שלנו ומשחקן היריב כדי לספור מיקומים שניתן להגיע אליהם באופן חוקי, ולקיחת המינימלי מביניהם (כי המשחק נגמר כאשר בפעם **הראשונה** שחקן לא מסוגל לנוע). משקלנו את הזמן הקצוב לתור גם לפי מספר התור כדי להגביר את הזמן הקצוב לתורות הראשונות של המשחק על חשבון התורות האחרונות כי ראינו אמפירית שזוהי אסטרטגייה מנצחת, ככל הנראה עקב היחס הגבוה בין השפעת התורות הראשונות של המשחק על מהלך המשחק לבין השפעת התורות האחרונים (בהם בדרך כלל פחות קריטי מה הצעד הנלקח כי מרחב האפשרויות כבר די קטן). ראוי לציין גם שלקראת סוף המשחק השפעת היוריסטיקה קטנה עקב הגעת האלגוריתם למצבי סוף, מה שגורם לוויתור על הערכת מצב עם יוריסטיקה בעד חישוב הערך האמיתי של מצב סוף. כיוון שתחת ההנחה שלנו חישובי יוריסטיקות דורשות את הרוב המוחלט של הזמן, הגעה למצבי סוף אלה גם מפחית לנו את הזמן הדרוש לטובת בחירת מהלך נבון, מה שגם מצדיק הפחתת זמן מתורות סוף המשחק.

**חלק ז**

**שאלה 18**

התוצאות שקיבלנו הן, שבריצת הסוכן AlphaBeta הגענו לעומק גדול יותר עם אותו זמן תור לעומת סוכן ה Minimax, כתוצאה מכך בלוחות סימטריים עבור השחקנים, נראה שהשחקן AlphaBeta לוקח צעדים יותר "הגיוניים", ויותר מנצח מאשר לא, וזה תואם לציפיות מכיוון שהגענו עמוק יותר בעומק העץ, מה שאמור לתת לנו יותר מידע עבור תורות עתידיים.

**שאלה 19**

הגדרנו 5 לוחות שונים וביצענו את הניסויים המוגדרים בשאלה:

ניסוי 1:

ניסוי 2 :

תוצאות הניסויים יחסית דומות, קיבלנו כי עבור עומק גדול יותר של מינימקס של שחקן ה LightABPlayer, הוא מצליח לצמצם פערים משחקים ה HeavyABPlayer, ואף לעבור אותו עם עומק גדול יותר.

בניסוי ה 2 השחקן עם ההיורסטיקה הפשוטה, LightABPlayer מצליח לנצח יותר טוב את HeavyABPlayer,לדעתנו מכיוון שלשחקן HeavyABPlayer יש עומק נמוך יותר ביחס ל LightABPlayer ביחס לניסוי הראשון.

עבור עומק זהה, כצפוי, קיבלנו כי השחקן HeavyABPlayer מצליח לנצח את LightABPlayer ברוב המשחקים.