תרגיל בית 2 : חיפוש רב סוכני
Reversi

מטרות התרגיל

- התנסות באלגוריתמים לחיפוש רב סוכני.
- מימוש אלגוריתם $Minmax$ ומספר שיפורים.
- התנסות בפיתוח יוריסטיקה למשחק Reversi.
- ביצוע מחקר המשווה ביצועים של שחקנים.

הערות

- תאריך הגשה : 1/1/18 23:59.
- את המטלה יש להגיש **בזוגות בלבד!**
- שאלות בנוגע לתרגיל יש לשלוח **למייל הקורסי** : ai.technion@gmail.com
- שאלות אחרות ובקשות לדחייה יש לשלוח **למיכל** : sbadian@cs.technion.ac.il
- קִראו היטב את ההסברים וההוראות במסמך זה, מטרתם לסייע לכם בהבנת הדרישות של התרגיל. שימו לב להוראות ההגשה המצורפות בסוף התרגיל.
- הִתְעַדְכְּנוּ ברשימת ה-FAQ באתר הקורס בתדירות גבוהה, לפני פנייה בשאלות דרך המייל ולפני הגשת התרגיל. ההערות שתתפרסמנה באתר הקורס מחייבות את כלל הסטודנטים בקורס!
- עקבו בתשומת לב רבה אחר הוראות ההגשה המצורפות במהלך התרגיל ובסופו לפני הגשתו.
- בתרגיל זה, כמו גם בתרגילים הבאים בקורס, הרצת הניסויים עשויה לקחת זמן ולכן מומלץ מאוד להימנע מדחיית העבודה על התרגיל לרגע האחרון. **לא תינתנה דחיות על רקע זה.**

מבוא והנחיות

- במטלה זו תתכננו ותממשו שחקנים למשחק $Reversi$ (Othello).
- את חוקי המשחק ניתן למצוא בוויקיפדיה.
- לפני שאתם ניגשים לתרגיל, מומלץ להתנסות מעט במשחק על-מנת לקבל הבנה טובה יותר שלו.
  ניתן למצוא גרסה חינמית online באינטרנט וכן מידע נוסף על אסטרטגיות ומאמרים.
- **<u>העתקות</u> יטופלו בחומרה בדין משמעתי.**
- מומלץ לחזור על שקפי ההרצאות והתרגולים בנושא "חיפוש רב סוכני" לפני תחילת העבודה על התרגיל.
- חבילת הקוד שאתם מקבלים מכילה את המימוש של המשחק וכן מימוש של שלושה שחקנים:
  שחקן אינטראקטיבי (המחכה לקלט מהמשתמש לשם ביצוע מהלך),
  שחקן רנדומלי המגריל צעד חוקי כלשהו
  ושחקן עם יוריסטיקה פשוטה.
  בנוסף, כל שחקן מחויב לשתי מגבלות זמן:

1. זמן אתחול (setup): מגבלה על זמן הריצה של הבנאי ($constructor$). בזמן זה יוכל השחקן להתכונן למשחק, לאתחל מבני נתונים וכו'. לצורך התרגיל (והתחרות) נקבע זמן זה להיות 2 שניות.
2. זמן עבור k מהלכים: מגבלה על הזמן הלוקח לשחקן לבחור את המהלך הבא. לכל k מהלכים במשחק יש אותה מגבלת זמן, כאשר על השחקן לבחור איך לחלק את הזמן הנ"ל בין k המהלכים.
   זמן שאינו מנוצל בתום k מהלכים אינו נצבר לטובת k המהלכים הבאים.
   בתרגיל (ובתחרות): $k = 5$.

בעוד שזמן האתחול הוא קבוע, הזמן עבור k מהלכים עשוי להשתנות בין משחקים, ולכן השחקנים שתממשו יפעלו במתכונת anytime-contract עם מדיניות ניהול זמנים לבחירתכם.

חלק א' – היכרות עם הקוד והמשחק

הורידו את חבילת הקוד שקיבלתם, הבינו את המבנה והמימוש של המשחק.

וודאו שהבנתם מהי היוריסטיקה הבסיסית של השחקן הפשוט וכיצד ה flow של האפליקציה עובד.

הריצו את השחקן הרנדומאלי מול השחקן הפשוט שש פעמים, שלוש פעמים השחקן הפשוט מתחיל ושלוש פעמים השחקן הרנדומאלי מתחיל. כתבו את התוצאות והסבירו אותן.

חלק ב' – בניית סוכן משופר (10 נק')

אתם נדרשים לממש שחקן מתוחכם יותר מהשחקן הפשוט. על השחקן להיות ממומש בתיקייה נפרדת על פי שמו $better\_player$. מחלקת השחקן תקרא $Player$ ועליה לרשת מהמחלקה $AbstractPlayer$.

I. הגדירו באופן מפורש יוריסטיקה משלכם להערכת מצבי המשחק (מחלקת שחקן, פונקציית ה $utility$). בפרט, יש לספק נוסחה עבור כל פרמטר של המשחק הנלקח בחשבון לשם חישוב היוריסטיקה. היוריסטיקה צריכה להכיל **לפחות** 3 פרמטרים חדשים על פני השחקן הבסיסי.

II. הסבירו את המוטיבציה להגדרה זו (ולכל פרמטר המופיע בחישובים שלכם). האם אתם צופים שהיא תשפר את ביצועי השחקן ביחס ליוריסטיקה של $simple\_player$?

III. ממשו שחקן בשם $better\_player$ הנעזר ביוריסטיקה המשופרת שהגדרתם, ללא תוספות נוספות.

IV. חזרו על ההרצה מחלק א, הפעם simple vs better, צרפו והסבירו את התוצאות שקיבלתם.

חלק ג' – בניית סוכן Min-Max (25 נק')

אתם נדרשים לממש שחקן המשתמש באלגוריתם min max. על השחקן להיות ממומש בתיקייה נפרדת על פי שמו min_max_player. מחלקת השחקן תקרא $Player$ ועליה לרשת מהמחלקה $AbstractPlayer$.

I. בנו את האלגוריתם כפי שנלמד בכיתה בקובץ utils.py. לצורך נוחות הוגדרה מחלקה ריקה בקובץ עם הפרמטרים הרלוונטיים באתחול וחתימת הפונ' search. אתם יכולים לבצע שינויים במחלקה.
- עליכם להתייחס למגבלת הזמנים של השחקן, יש <u>לעצור</u> את החיפוש של האלגוריתם במקרה שהזמן הסתיים (שדה: $self.no\_more\_time$).
- הפונ' search מקבלת פרמטר המציין את עומק החיפוש בעץ, עליכם לממש את האלגוריתם <u>בשיטת העמקה הדרגתית.</u>

II. ממשו את השחקן בשם min_max_player כאשר:

- היוריסטיקה (פונ' הutility במחלקת שחקן) היא לבחירתכם, ניתן להשתמש בזו של $simple\_player$ או בזו מחלק ב'.
- הקריאה לאלגוריתם $Min - Max$ נעשית מתוך הפונ' get move במחלקת השחקן.

- בפונ' get move בצעו איטרציות עם המשתנה depth והאלגוריתם שכתבתם. כל עוד יש לשחקן זמן עליו להעמיק בעץ כדי לחפש מהלך טוב יותר.

חלק ד' – בניית סוכן $Alpha - beta$ (10 נק')

אתם נדרשים לשפר את יעילות השחקן מחלק ג' בעזרת אלגוריתם $alpha\_beta\_pruning$. על השחקן להיות ממומש בתיקייה נפרדת על פי שמו $alpha\_beta\_player$. מחלקת השחקן תקרא $Player$ ועליה לרשת מהמחלקה $AbstractPlayer$. <u>הדגשים בסעיף זה זהים לאלו מחלק ג'</u>.

I. בנו את האלגוריתם כפי שנלמד בכיתה בקובץ utils.py. לצורך נוחות המימוש הוגדרה מחלקה ריקה בקובץ עם הפרמטרים הרלוונטיים באתחול וחתימת הפונ' search.

II. ממשו את השחקן בדומה לחלק ג', אך כעת עם הקריאה לאלגוריתם המשופר.

III. האם אתם מצפים שיהיה הבדל בין ביצועי שחקן min-max לשחקן alpha-beta? באילו תנאים?

חלק ה' – תיאוריה ושיפורים (5 נק')

ענו על השאלות הבאות:

I. מתוך כל הסוכנים שראיתם עד כה, מאיזה שחקן אתם מצפים לביצועים הטובים ביותר ומדוע.

II. עבור הפונקציות הבאות, ענו -
איזה שיפורים נוספים ניתן היה לקבל דרכן.
איך היה נראה המימוש שלכם – יש לתאר תכנון אך אין צורך לפרט ברמת מבני הנתונים.
1) פונקציה selective deepening מחוברת לאלגוריתם min max אשר מאפשרת ניהול החלטה מתי להמשיך לבצע להעמקה בעץ.
2) פונקציה time for step לניהול זמנים עבור השחקן – חלוקת זמן שונה ולא אחידה בין k מהלכים.

חלק ו' – שימוש במאגר משחקים (10 נק')

I. קראו על תוכנית logistello. תארו אפשרויות ואסטרטגיות נוסף על מה שמימשתם בחלקים הקודמים, שהתוכנה משתמשת כדי לנצח את המשחק.

II. כחלק מחבילת הקוד שקיבלתם, מופיע הקובץ $book.gam$ אשר היה בשימוש בתוכנת logistello. קובץ זה הוא בפורמט של ascii וכל שורה בו מהווה **משחק מלא ותוצאה סופית**.

הלוח אליו מתייחסות השורות נראה כך: (עמודות באותיות קטנות ושורות במספרים החל מ1).

	a	b
1		
2		

לנוחיותכם תוכלו לראות את המשחק הראשון בקובץ בתמונה שמצורפת לקוד.

1. הסבירו מהו opening book, מהי התוספת והיתרונות שהוא נותן לשחקן.
2. יש ליצור opening book מתוך קובץ המשחקים לפי המתודולוגיה הבאה:

   עליכם לחלץ מהקובץ שקיבלתם 70 פתיחות (=10 מהלכים ראשונים, 5 שחקן - 5 יריב) של המשחקים הנפוצים מתוך הקובץ ולדרג אותם לפי פופולריות.

   מהם 5 המשחקים הראשונים במילון שלכם? כתבו אותם ביחד עם הדירוג שלהם.
3. צרו פונ' בשחקן $better\ player$, בשם $opening\_move$ אשר מנסה לבחור את המהלך הבא ע"י חיפוש במילון שיצרתם ב O(1). המהלך n ייבחר בהינתן ש prefix של n-1 מהלכים קודמים לו הופיעו במשחק.

   במידה ולא נמצאה פתיחה רלוונטית במילון בחירת המהלך תהיה לפי קוד המקור של $better\_player$.
4. מה החיסרון בבניית ספר פתיחות בגישה שתוארה?
5. חשבו על עוד דרכים להשתמש בקובץ המשחקים (לחלופין תוכלו להציע שיפור של הגישה שהוצגה).

   כתבו לנו מה ה design שלכם.
6. אופציונלי: אתם יכולים להשתמש בקובץ כדי לשפר את השחקן שאתם שולחים לתחרות הקורסית.

חלק ז' – ניסויים, תוצאות ומסקנות (40 נק')

נרצה להשוות בין השחקנים הנעזרים ביוריסטיקות השונות שראינו, עם ובלי שיפורים, תחת מגבלות זמן שונות. בפועל, ההשוואה תיעשה בין 4 שחקנים:

(1) $simple\_player$ המסופק

(2) $better\_player$

(3) min_max_player

(4) $alpha\_beta\_player$

נסמן $T = \{2,10,50\}$. עבור כל זוג שחקנים (סה"כ $\binom{4}{2}$ צמדים) הריצו סדרת משחקים עבור כל מגבלת זמן $t \in T$ (בשניות). להזכירכם, t מגביל את זמן החישוב של השחקן עבור k מהלכים (כפי שהוסבר במבוא). בכל מגבלה של t, הריצו 5 משחקים כפול 2 ורסיות - אחת לכל חלוקת O\X בין השחקנים,

סה"כ $3_{t\in T} \cdot \binom{4}{2} \cdot 2 \cdot 5$ משחקים.

לצורך ניתוח התוצאות, נשתמש בשיטת הניקוד הבאה:

- הניקוד למשחק יחיד - ניצחון = 1 נק', תיקו = 0.5 נק', הפסד = 0 נק'.
- הניקוד הכולל של שחקן יחושב כסכום הנקודות שלו מכל המשחקים בהם הוא לקח חלק.

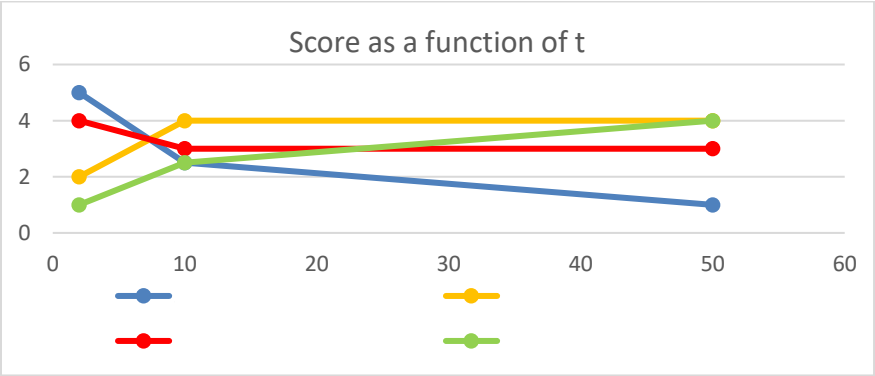
1. יש להגיש קובץ בשם $experiments.csv$ שיכיל את תוצאות המשחקים שהרצתם. כל שורה בקובץ תכיל 5 עמודות: שם שחקן X (מבין 4 השחקנים המשתתפים בניסוי), שם השחקן O, מגבלת הזמן בשניות ל$k = 5$ תורות, ניקוד השחקן X, ניקוד השחקן O.
לנוחיותכם מסופקת שורה לדוגמא:

$$simple\_player, better\_player, 10,0.5,0.5$$

אין לכלול כותרות בקובץ, אלא רק את תוצאות הניסויים.

2. הצגת התוצאות:

I. הציגו גרף המתאר עבור כל אחת ממגבלות הזמנים את <u>הניקוד הכולל של כל שחקן</u> כפונקציה של מגבלת הזמן. הקפידו להשתמש בצבעים או בסוגי קו שונים עבור שחקנים שונים. לנוחיותכם מצורף גרף לדוגמא $(*)$.



II. צרפו את טבלת הנתונים שיצרו אותו. למשל$(*)$:

$t = 50$	$t = 10$	$t = 2$	
1	2.5	5	$simple\_player$
4	4	2	$better\_player$
3	3	4	min_max_player
4	2.5	1	$alpha\_beta\_player$

$(*)$ שימו לב כי הגרפים והטבלאות שהצגנו כאן נוצרו לצורך **המחשה בלבד**.

III. השוו את ביצועי השחקנים ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה בניקוד הכולל של כל אחד מהשחקנים. האם הגרף ענה על הציפיות שלכם? באילו מקרים היוריסטיקה שלכם מתגברת על היוריסטיקה הפשוטה? מהן נקודות הקיצון בגרף (אם יש)? הסבירו <u>בהרחבה</u> את כל המאפיינים שראיתם ובצעו ניתוח מעמיק בין **כל** זוג שחקנים. (10 נק').

חלק ח' – תחרות בקורס

כל זוג נדרש לממש שחקן יחיד שייצג אותו במשחק Reversi הקורסי. יתקיים טורניר בין השחקנים שיוגשו, והסטודנטים ששחקניהם יצטיינו בתחרות יזכו בבונוס לציון הסופי של הקורס.

<u>בחירת השחקן:</u>

אתם רשאים להשתמש באחד השחקנים שמימשתם בסעיפים הקודמים, לשפר או לממש שחקן חדש כרצונכם, למטלה זו אין משקל בניקוד תרגיל בית 3 עם סייג אחד: **במידה והשחקן שתגישו לא ירוץ, תהיה הורדת נקודות על התרגיל. אין להגיש את שחקן ה simple שקיבלתם.**

<u>מסגרת הזמן:</u>

בתחרות שנקיים בין השחקנים שתגישו, יאכפו שתי מגבלות זמנים:

- מגבלת זמן של 2 שניות לשחקן עבור שלב האתחול.
- מגבלת זמן של מספר שניות לשחקן עבור כל 5 מהלכים.

מגבלות אלו זהות לאלו שבחנתם במהלך התרגיל. הסוכנים ייבחנו תחת 3 מגבלות הזמנים שהוגדרו בשלב הניסוי. זמן החישוב הכולל של כל k מהלכים צריך להיות נמוך מהמגבלה המוגדרת. שחקן שיחרוג מהזמן המוקצה, יפסיד טכנית במשחק הספציפי בו קרתה החריגה.

<u>חוקים:</u>

- מימוש השחקן יופיע בתיקיה בשם $competition\_player$. על מחלקת השחקן להקרא $Player$ ולרשת מהמחלקה $AbstractPlayer$.
- אסור לשחקן להשתמש ברשת האינטרנט או ברשת מקומית כלשהי.
- אסור לשחקן להשתמש בכל סוג של קוד מקבילי.
- צריכת הזיכרון של השחקן שלכם מוגבלת ל-512 MB.
- כל שחקן שינסה לרמות או לשבש את קוד היריב או המערכת ייפסל.
- חל איסור להשתמש במידע שעובד מראש ($pre-processed$). ניתן לבצע פעולות אתחול ו/או עיבוד מקדים בבנאי (__init__) של השחקן בלבד, וזאת במסגרת מגבלת זמן האתחול.
- אין להשתמש ב-C extensions.

<u>פרסים:</u>

מקום ראשון: 3 נק'.

מקום שני: 2 נק'.

מקום שלישי: 1 נק'.

הוראות הגשה

- הגשת התרגיל תתבצע **אלקטרונית בלבד** דרך אתר הקורס.
- כהכנה להגשה, אתם מתבקשים ליצור בתוך התיקייה $players$ שבקוד המסופק **תיקיה יחידה** בשם $AI2\_\langle id1\rangle\_\langle id2\rangle$ (ללא הסוגריים המשולשים), שבתוכה הקבצים הבאים:
  - קובץ בשם $readme.txt$ בפורמט הבא:

name1 id1 email1

name2 id2 email2

  - קובץ ריק בשם $\_\_init\_\_.py$
  - הקובץ utils.py.
  - 4 תיקיות ששמן:
$better\_player, min\_max\_player, alpha\_beta\_player, competition\_player$
עבור 3 השחקנים שמימשתם בחלק ב'-ד' והשחקן אותו אתם מגישים לתחרות.
על כל אחת מהתיקיות הללו להכיל קובץ $\_\_init\_\_.py$ עם קוד השחקן שלכם (או import לקוד).
  - כל קובץ עזר שכתבתם, אשר קוד השחקנים משתמש בו.
  - קובץ בשם $requirements.txt$ שיכיל כל חבילה חיצונית שאינה מותקנת כחלק מ-$Anaconda\ Python$. על אחריותכם לוודא כי ניתן להתקין כל חבילה כנ"ל באמצעות הרצת השורה: $pip\ install -r\ requirements.txt$
  - קובץ בשם $AI\_HW2.pdf$, המכיל את התשובות לחלק היבש והערות מיוחדות הנוגעות לקוד במידה ויש צורך בכך.

- לאחר החלפת הקובץ utils, בדקו שאתם מסוגלים להריץ את הפקודה הבאה:

$$python\ run\_game.py\ 2\ 3\ 5\ y\ AI2\_\langle id1\rangle\_\langle id2\rangle.competition\_player\ simple\_player$$

פקודה זו תריץ משחק בין שחקן התחרות שלכם (בתור השחקן X) לבין השחקן הפשוט.

- את התיקייה $AI2\_\langle id1\rangle\_\langle id2\rangle$ יש לקבץ לארכיון בשם $AI2\_\langle id1\rangle\_\langle id2\rangle.zip$, **שאותו (ואותו בלבד) עליכם להגיש.**
- אין צורך להעתיק את הקבצים המסופקים לכם אל תוך תיקיית ההגשה. הניחו כי קבצים אלו יהיו זמינים בעת בדיקת התרגיל.
- שימו לב שכל הפנייה למיקום קובץ/תיקייה כלשהם בקוד תהיה רלטיבית ($relative\ path$) ולא אבסולוטית, כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקיית הפרויקט. הקפידו לבדוק זאת לפני ההגשה!
- הקפידו על קוד **<u>ברור, קריא ומתועד!</u>** עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם.
- "המצאת" נתונים לצורך בניית הגרפים **אסורה** ותוביל לדיון בבית הדין המשמעתי של הטכניון.

בהצלחה!