דו"ח מסכם
תרגיל בית 2

יונתן בתן

302279138

Yonibettan@gmail.com

נדב אליהו

303086854

Neliahu@gamil.com

חלק א':

מטעמי אסטטיקה, הודבקו התוצאות עבור ריצת התכנית ללא הדפסות.

```
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n simple_player random_player
The winner is O random
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n simple_player random_player
The winner is X simple
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n simple_player random_player
The winner is X simple
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n random_player simple_player
The winner is O simple
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n random_player simple_player
The winner is O simple
(py3) ybettan@localhost hw2 (devel) $ python run_game.py 2 10 5 n random_player simple_player
The winner is O simple
(py3) ybettan@localhost hw2 (devel) $
```

לפי התכנית השחקן שמוכנס ראשון (מבין השניים) הינו X.

ניתן להבחין כי ב- 3 ההרצות הראשונות כיוון ש- simple הוא הראשון שהוכנס אזי הוא מקבל את התווית X ו-random את התווית O.

ב-3 הריצות האחרונות נראה כי simple הוא שקיבל את התווית O

השחקן X תמיד משחק ראשון ולכן לצורך ההרצות פשוט שינינו את סדר הפרמטרים בשורת הפקודה.

הפלט מייצג את השחקן המנצח בריצה הספציפית.

התוצאות הינן שונות בין הריצות מכיוון שהשחקן random תמיד מגריל מצב אקראי מבין המצבים האפשריים העומדים לרשותו.

חלק ב':

התשובות לא לפי סדר הסעיפים לצורך נוחות ההסבר אך הם עונים על כולם.

אנו נבחר יוריסטיקה אשר לה 2 מרכיבים עיקריים:

- **תנועה:**
  - בוחנת לכל משתתף את מספר המהלכים האפשריים
  - בודקת מתוכם כמה מהלכים הם בעלי פוטנציאל לכל משתתף, כלומר משבצות אשר בקרבתן יש חייל יריב אחד לפחות
- **ניקוד:**
  - מבוסס על מספר החיילים שיש לכל משתתף
  - ניתן דגש על פינות ופאות
  - מתוך הפאות ניתן דגש על החיילים בפאות שהינם במצב "יציב" כלומר לא ניתנים להפיכה (למשל חלק משורה המסתיימת בפינה)

ולבסוף נשלב את 2 מרכיבים אלו בעזרת משקלים שונים אשר משתנים לפי שלבי התכנית, כלומר בהתחלה ניתן דגש גדול יותר על תנועה שגם התנועה משמעותית בהתחלה וגם אנחנו רחוקים מהפינות ובשלב כלשהו נתחיל להוריד את המשקל של התנועה לטובת הניקוד שכן אנו מתקרבים לסוף המשחק והתנועה יורדת משמעותית ל-2 השחקנים.

עבור מצב סופי היוריסטיקה תחזיר את הפרש משפר החיילים שלנו ושל היריב.

אנחנו אכן צופים שיפור משמעותי בתוצאות שכן היוריסטיקה הינה מיודעת יותר.

בלוח 8X8 כמו אצלנו יש 64 משבצות שמתוכם 4 כבר תפוסות במצב התחלתי, מה שמשאיר אותנו עם 60 מהלכים .

- ב- 20 המהלכים הראשונים ניתן פאקטור 1.5 לטובת התנועה ו-0.5 לטובת הניקוד
- ב- 20 המהלכים האמצעיים ניתן פאקטור 1 לשני המרכיבים
- ב- 20 המהלכים האחרונים ניתן פאקטור 0.5 לטובת התנועה ו-1.5 לטובת הניקוד

ועכשיו קצת נוסחאות:

<u>ניקוד:</u>

$$\begin{aligned} result = &(\#myCorners - \#oponentCorners) \cdot CORNER\_FAC + \\ &(\#myStableSides - \#oponentStableSides) \cdot STABLE\_EDGE\_FAC + \\ &(\#mySides - \#oponentSides) \cdot EDGE\_FAC + \\ &(\#myInter - \#oponentInter) \cdot INTER\_FAC \end{aligned}$$

בשלב זה ניתן את הערך הנ"ל לקבועים אך יתכן ונשנה אותם בעתיד לצורך שיפור ביצועים לתחרות

$$CORNER\_FAC = 100$$
$$STABLE\_EDGE\_FAC = 20$$
$$EDGE\_FAC = 5$$
$$INTER\_FAC = 1$$

<u>תנועה:</u>

$$result = (\#myLegalMoves - \#oponentLegalMoves) \cdot MOBILITY\_FAC$$

בשלב זה ניתן את הערך הנ"ל לקבועים אך יתכן ונשנה אותם בעתיד לצורך שיפור ביצועים לתחרות

$$MOBILITY\_FAC = 10$$

<u>מצב סופי:</u>

עבור מצב סופי נחזיר את $result = \#myInter - \#oponentInter$

חלק ג' - סעיף 2:

ישנם k אנשים שלכל אחד מהם יש הזמנה לכן אחד ממקרי הקצה הוא כאשר כל הלקוחות עלו ואף אחד עוד לא ירד (ישנם עוד מקרים עם מקדם הסתעפות זהה) ולכן יש k תחנות אופציונליות.

אחרי שאחרון הלקוחות ירד מקדם הסיעוף הוא 0 (אם לא מתחשבים במצב מטרה לצורך מקדם הסיעוף אזי מקדם הסיעוף המינימלי יהיה 1)

לסיכום
$$\max = k$$
$$\min = 0$$

סעיף 3:

לא יתכנו מעגלים.

מכיוון שאיחוד הרשימות נותן את רשימת כל ההזמנות ונתון שבכל מצב מובטח שלפחות אחת הרשימות השתנתה (רשימה משתנה רק בכיוון יחיד $W \to B \to F$) לכן לא קיימת האפשרות להגיע לאותו מצב יותר מפעם אחת.

סעיף 4:

מילולית:

קבוצת כל המצבים בהם כל ההזמנות גמורות ואין הזמנות באוטובוס או הזמנות ממתינות והצומת היא צומת ירידה של הזמנה כלשהי.

פורמלית:

$$G = \{ \ s = (v, W, B, F) \in S \mid s = (t_i \in V, \Phi, \Phi, Ord) \ , 1 \le i \le k \ \}$$

סעיף 5:

מצב מטרה הוא מצב בו הלקוח האחרון יורד מהאוטובוס. מכיוון שישנם k יעדים של לקוחות אזי ישנם k מצבי מטרה בהנחה וכל מצבי המטרה שונים.

סעיף 6:

לא יתכנו בורות.

כל עוד ישנן הזמנות לא גמורות אופרטור המעבר מאפשר לנו לעבור למצב בו צריך להוריד\לאסוף לקוח

סעיף 7:

$$Succ\left(\left(v_1, W_1, B_1, F_1\right)\right) = \{ \ \underbrace{\left(s_{\bar{i}}, W_1 - \bar{i}, B_1 + \bar{i}, F_1\right)}_{taking\ \ passanger} \vee \underbrace{\left(t_{\bar{i}}, W_1, B_1 - \bar{i}, F_1 + \bar{i}\right)}_{dropping\ \ passanger} \ , \ \bar{i} = \{i_{n_1}, i_{n_2}, ... i_{n_r}\} \ , \ 1 \le n_t \le k \ \}$$

- מכיוון שיתכן שנאסוף\נוריד יותר מלקוח אחד בצומת מסוים אזי $\bar{i}$ מייצג את כל ההזמנות המטופלות באותו צומת
- במידה וההנחה של סעיף 5 תקפה גם כאן נקבל $\bar{i} \equiv i$

סעיף 8:

לפי ההנחה בכדי לטפל ב- k הזמנות יש לעבור ב- $2k$ צמתים ולכן עומק מצב מטרה במרחב החיפוש הינו $2k$.

- המינימלי והמקסימלי מזדהים בבעיה זו

חלק ד' - סעיף 9:

.a

```
(py3) ybettan@localhost AI1 (devel) $ python scriptsAndExperiments/initial.py
load_map_from_csv: 14.73sec
One of the orders is from junction #23695 at (32.1022894, 34.9882995) to #33320 at (32.0926573, 35.1022635)
A lower bound on the distance we need to drive for this order is: 10.78km
```

.b

```
load_map_from_csv: 28.71sec
One of the orders is from junction #23695 at (32.1022894, 34.9882995) to #33320 at (32.0926573, 35.1022635)
A lower bound on the distance we need to drive for this order is: 10.78km
Path length: 161.11km

Process finished with exit code 0
```

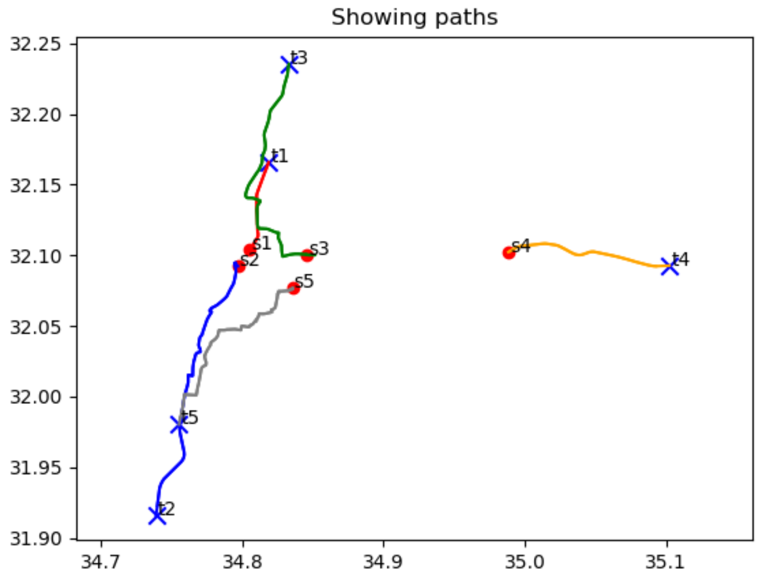
.c תחילה הקוד טוען את מפת הדרכים ואת פרטי ההזמנות למשתנים roads, prob. לאחר מכן נבחרת הזמנה באקראי ומודפס המרחק האווירי שלה, כלומר המרחק האווירי בין המקור ליעד של ההזמנה.
לבסוף מייצרים מסלול אשר עובר בכל הצמתים המתקבלים כקלט (רשימה ארוכה hard coded) ומחשבים את אורכו

- הפלט המתוקן הוכנס לסעיף קודם

סעיף 10:

```
Junction idx: #851288
waiting for bus: [(23695, 33320)]
orders on bus: [(32056, 834603), (851288, 533396)]
finished orders: [(47521, 606430), (466524, 29249)]
Is goal? False
```

חלק ה' - סעיף 11:




```
Shortest distance for order from #32056 to #834603: 7.15km
Shortest distance for order from #47521 to #606430: 22.34km
Shortest distance for order from #466524 to #29249: 20.30km
Shortest distance for order from #23695 to #33320: 11.28km
Shortest distance for order from #851288 to #533396: 16.28km
Total distance: 77.35km
```

סעיף 12:

נבחר בטענה C – לא ניתן לקבוע באופן כללי, הקוד מחשב את המסלול האופטימלי באמצעות A* עבור כל הזמנה בנפרד, כלומר לא מתחשב בשילוב ההזמנות – לכן במקרים מסוימים סכום הדרכים יהיה סכום הכולל דרכים חופפות (כמו שניתן לראות בין המסלולים בצד שמאל) ובכך בעצם להגדיל את אורך המסלול שיכל להיות קצר יותר, ובמקרים אחרים יכול לכלול מסלול מנותק שעל מנת לחבר אותו לשאר ההזמנות דורש נסיעה אליו – כלומר חסר חלק במסלול (כמו המסלול הימני בגרף).

חלק ו' - סעיף 13:

TLV_5	135.06km
SDEROT_50	539.59km
BEER_SHEVA_100	565.68km
HAIFA_100	1329.21km

סעיף 14:

מאחר ובאלגוריתם אנו עוברים על כל הצמתים הנוספים שיש בהם הזמנות (ירידה או עליה) ואנו מתבססים על ההנחה כי בכל צומת יש רק לקוח אחד שיורד או עולה – כדי לתת שירות לכולם נצטרך לפתח את כלל הצמתים (2k) כאשר הצומת האחרון לא דורש פיתוח כי נגיע למצב מטרה. נשים לב שזה החסם התחתון אך גם העליון מאחר ולא יתכן ונעבור בצומת ללא קשר ללקוח.

$$\max = \mathbf{2}\boldsymbol{k} - \mathbf{1} = min$$

סעיף 15:

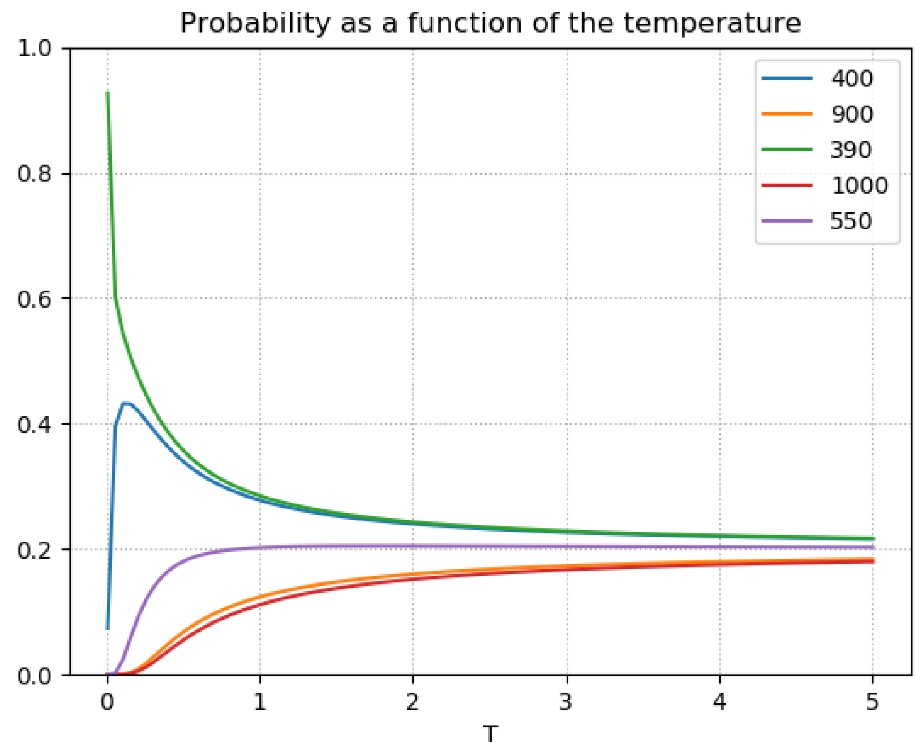
סיבוכיות המקום על פי סעיף 14 תהיה $\boldsymbol{\theta(k)}$

חלק ז' - סעיף 16:

ניתן לראות כי הסקאלה לא משפיעה על ההתפלגות מאחר ומדובר בכפל וחילוק של אותו קבוע

$$\Pr(pnt_i) = \frac{\left(\frac{x_i}{\alpha}\right)^{-\frac{1}{T}}}{\sum_{pnt_h}\left(\frac{x_h}{\alpha}\right)^{-\frac{1}{T}}} = \frac{\cancel{\alpha^{\frac{1}{T}}}(x_i)^{-\frac{1}{T}}}{\cancel{\alpha^{\frac{1}{T}}}\sum_{pnt_h}(x_h)^{-\frac{1}{T}}} = \frac{(x_i)^{-\frac{1}{T}}}{\sum_{pnt_h}(x_h)^{-\frac{1}{T}}}$$

סעיף 17:



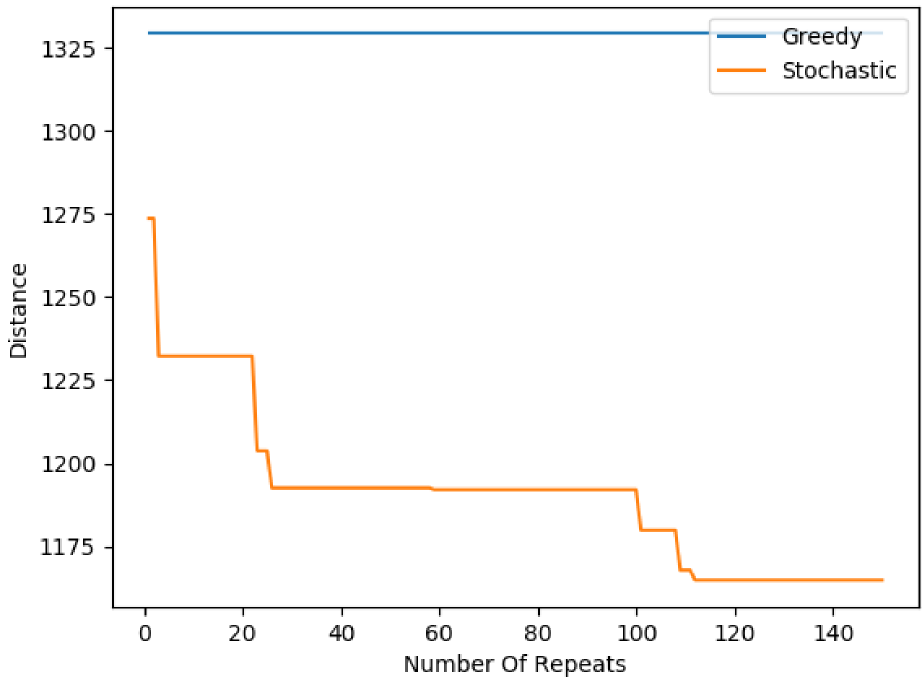
סעיף 18:

כאשר T שואף לאפס הבחירה שלנו תהיה דומה לבחירה גרידית רגילה ובה נבחר בSCORE הקטן ביותר.

סעיף 19:

כאשר T שואף לאינסוף אין משמעות לערכי הSCORE מאחר וה"רעש" הוא הדומיננטי ונקבל הסתברות שווה לכל x (חמישית)

סעיף 20:



סעיף 21:

average: 1288.9264199999993

variance: 55.70944063253793

p-value: 9.415679334520726e-17

אנו שמים לב כי p-value מאוד קטן ומכאן שההסתברות שהשערת האפס נכונה מאוד נמוכה לכן נדחה את תוצאת הניסוי.

מהתבוננות בנתונים ניתן להבחין כי התוחלת רחוקה מתוצאות החיפוש החמדן וערך סטיית התקן גבוהה

חלק ח' - סעיף 22:

ישנן 5 הזמנות מכאן שישנם 5 מצבי מטרה, אנו נרצה לעבור ב5 נק' איסוף ו5 נק' הורדה שאחת מהן תהיה מצב מטרה.

כל הזמנה מתוך ה5 צריכה להיות לפני ההורדה של אותה ההזמנה – כלומר יש לנו זוגות אשר הסדר ביניהם צריך להישמר.

כלומר $\frac{10!}{2^5}$ (האופציה לבחור 5 נק' איסוף ו5 נק' הורדה חלקי הסדר בין 5 הזוגות)

סעיף 23:

Greedy result: 135.06km

Stochastic (200 repetitions): 127.291km

A* (null heuristic): g(G)=127.29km, h(I)=0.00km, developed: 1640 states

סעיף 24:

תשובה האם מיודעת או לא התבצעה על פי ההגדרה בתרגול

	קבילה	מיודעת מ	נימוק קצר
a	כן		מחזירה את המרחק האווירי המקסימלי בין צומת מקור ליעד בין כלל הממתינים, מאחר והמסלול האמיתי יהיה ארוך יותר בין 2 הנק' ובדרך אולי עוד נעבור בעוד נק' הרי שזה קביל
b	כן	a	מחזירה את המרחק האווירי המקסימלי בין כלל צמתי המקור לכלל צמתי היעד, מאחר ונעבור בכלל הצמתים הרי שלא יתכן שנעבור מרחק קצר יותר מהמרחק האווירי הגדול ביותר ולכן קביל.

			מיודעת לa מאחר ומכילה את כלל האופציות של a בתוספת של עוד כלומר הערך המקסימלי רק יכול לגדול או להישאר זהה
c	לא		מחזירה את המרחק האווירי המקסימלי בין צומת מקור לצומת היעד של נוסעי האוטובוס לא קביל מאחר ובצומת המקור שלהם כבר ביקרנו ולכן חלק מהדרך בוצעה ויתכן כי המרחק הנותר קטן יותר מהמרחק האווירי
d	כן		מחזירה את המרחק האווירי המקסימלי בין הצומת הנוכחית לכלל צמתי המקור של הממתינים, מאחר ונצטרך לבקר בכל הצמתים האלו מן הסתם שהדרך תהיה ארוכה יותר מהמרחק האווירי (או שווה לו) לכן קבילה
e	כן	a, d	מחזירה את המרחק האווירי המקסימלי בין כל הממתינים ליעד שלהם פלוס המרחק האווירי המקסימלי מהצומת הנוכחית לכלל צמתי האיסוף, מאחר וכדי לאסוף נוסע נצטרך לנסוע מהצומת הנוכחית לצומת האיסוף שלו ובסופו של דבר להגיע לצומת ההורדה הרי שנעבור במסלולים לצמתים אלו שיהיו ארוכים מהמרחק האווירי ולכן קבילה, מכילה בתוכה את a וd לכן מן הסתם הערך המקסימלי יהיה כאן לפחות כמו בa וd ומכן שהיא מיודעת אליהן.
f	כן	a	מחזירה את המרחק האמיתי המקסימלי בין צמתי המקור לצמתי היעד של כלל הממתינים, מאחר ואנו נהיה חייבים לאסוף את כולם ולפזר אותם הרי שנעבור במסלול האמיתי המיטבי של כל הצמתים האלו או אפילו נאריך אותו בדרך לאיסופים נוספים לכן קבילה. המרחק האמיתי תמיד גדול או שווה לאווירי ולכן מיודעת לa
g	כן	f, b	מחזירה את המרחק האמיתי המקסימלי בין צומת מקור של ממתין כלשהו לצומת הורדה ממתין כלשהו, מאחר ונצטרך לאסוף את כלל הממתינים ולהורידם הרי שנעבור בכלל הצמתים לאיסוף וכלל הצמתים להורדה ומכאן שהמרחק האמיתי יכול להיות רק גדול מהמקסימלי בין 2 מהם. לכן קבילה, מרחק אמיתי תמיד גדול או שווה למרחק אווירי לכן היא מיודעת לb,a בנוסף האופציות מקסימום שלה מכילות את האופציות של f ולכן בוודאות גדולה או שווה לf ומכאן גם מיודעת לf

סעיף 25:

Greedy result: 135.06km

Stochastic (200 repetitions): 127.291km

A* (null heuristic): g(G)=127.29km, h(I)=0.00km, developed: 1640 states

A* (Custom heuristic): g(G)=127.29km, h(I)=20.37km, developed: 1672 states

סעיף 26:

Greedy result: 135.06km

Stochastic (200 repetitions): 127.291km

A* (null heuristic): g(G)=127.29km, h(I)=0.00km, developed: 1640 states

A* (Custom heuristic): g(G)=127.29km, h(I)=20.37km, developed: 1672 states

MST heuristic is computing required metadata...

A* (MST heuristic): g(G)=127.29km, h(I)=81.34km, developed: 475 states

פרק שני - סעיף 27:

א.

נתון כי h קבילה, מכאן שעבור כל מצב s בו h מוגדרת $\mathrm{h_0(s)} = \mathrm{h(s)}$ ולכן $\mathrm{h_0(s)} < h^*(s)$, עבור מצבים s בהם h אינה מוגדרת $h_0(s) = 0 \le h^*$ לכן $0 \le h_0(s) \le h^*$ לכל s ולכן קבילה.

ב.

נגדיר את היורסטיקה הבאה:

$$h_{tree}(s) = \begin{cases} h(s), & Applicable_h(s)\ is\ True \\ 0, & elseif\ s = root\ and\ Applicable_h(s)\ False \\ \max(father.h - cost(father, s)\ ,\quad 0)\ , & else \end{cases}$$

ולמעשה נריץ עם יוריסטיקה זו *A ונקבל את התוצאה הרצויה.

ג.

נגדיר יוריסטיקה דומה לסעיף ב רק שהפעם אין לנו עץ לכן לא נדע את האבא של מצב ההתחלה ולכן אם h(s) לא מוגדרת נגדיר את h להיות 0. ובנוסף כאשר נגיע למצב בו כבר מחושבת יוריסטיקה לא נעדכנה שוב, כלומר:

$$h_{new}(s) = \begin{cases} h(s), & Applicable_h(s)\ is\ True \\ 0, & elseif\ s.father\ unknowen\ and\ Applicable_h(s)\ False \\ \max(father.h - cost(father, s)\ ,\quad 0)\ , & else \end{cases}$$

אבל האלגוריתם *A יחשב יוריסטיקה רק פעם אחת לכל מצב, נשים לב שבמקרה של מעגל אינסופי גם ביוריסטיקה המקורית לא היה פתרון לכן המצב טוב יותר או זהה.

ד.

קיים אלגוריתם כזה, נשים לב כי נתון לנו מרחב מצבים שהוא עץ כאשר לשורש יש יוריסטיקה מושלמת.

מאחר והיא מושלמת בכל צעד נוכל לחשב לכל בן של השורש את היורסטיקה המושלמת המשוערכת שלו שהיא היורסיטיקה של האב פחות המרחק לבן, נשים לב שיתכן שמבן זה אי אפשר להגיע למצב מטרה אבל נזכיר שללא הגדרה זו כל הבנים היו מקבלים ערך יוריסטי אפס ולכן עדיין חל שיפור במצב.

במקרה ונגיע לחישוב ערך יוריסטי שלילי נדע מידית לפסול את הכיוון המדובר ובכך לחסוך פיתוח צמתים. פעולה זו לבדה כבר משפרת את אלגוריתם *A.

כך נמשיך לכל מצב שנפתח – מאחר ובכל מצב תהיה לנו יוריסטיקה מושלמת משוערכת הרי שתמיד נבחר ביעד במסלול הטוב ביותר ולכן או שנבחר במסלול שהוא מבוי סתום (אבל היינו בוחרים בו בכל מקרה עם היוריסטיקה המקורית) – כלומר אנחנו רק משפרים את ביצועי *A.