יבש – תכנון ו – MDP

חלק 1:

בתרגול ראינו את משוואת בלמן כאשר התגמול ניתן עבור המצב הנוכחי בלבד, כלומר $\mathbf{R}: \mathbf{S} \rightarrow \mathbb{R}$, למתן תגמול זה נקרא "תגמול על הצמתים" מכיוון שהוא תלוי בצומת שהסוכן נמצא בו.

בהתאם להגדרה זו הצגנו בתרגול את האלגוריתמים Value iteration ו-Policy Iteration למציאת המדיניות האופטימלית.

כעת, נרחיב את ההגדרה הזו, לתגמול המקבל את המצב הנוכחי והפעולה לביצוע שבה בחר הסוכן, כלומר: $\mathbf{R}: \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$, למתן תגמול זה נקרא "תגמול על פעולה".

א. (2 נק') התאימו את הנוסחה של התוחלת של התועלת מהתרגול, עבור התוחלת של התועלת המתקבלת במקרה של "תגמול על פעולה", אין צורך לנמק.

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R\big(S_t, \pi(S_t)\big) \mid S_0 = s\right]$$

ב. (2 נק') כתבו מחדש את נוסחת משוואת בלמן עבור המקרה של "תגמול על פעולה", אין צורך לנמק.

$$U(s) = \max_{a \in A(s)} \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) * U(s') \right]$$

ג. (4 נק') נסחו את אלגוריתם Value Iteration עבור המקרה של "תגמול על פעולה".

התייחסו גם למקרה בו $\boldsymbol{\gamma} = \mathbf{1}$, והסבירו מה לדעתכם התנאים שצריכים להתקיים על הסביבה**mdp** על מנת שתמיד נצליח למצוא את המדיניות האופטימלית.

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
      rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
      δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow \max_{a \in A(s)} [R(s,a) + \gamma \sum_{s'} P(s'|s,a) * U(s')]$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

אם $\gamma = 1$: תנאי העצירה שך value_iteration יהיה כאשר $\delta = 0$, כלומר כאשר אין הפרש בין ערכי התועלת באטירציה הקודמת ובאיטרציה הנוכחית, במקרה זה תובטח התכנסות למדיניות אופטימלית כפי שלמדנו בקורס.

ד. (4 נק') נסחו את אלגוריתם Policy Iteration עבור המקרה של "תגמול על פעולה".
התייחסו גם למקרה בו $\gamma = 1$, והסבירו מה לדעתכם התנאים שצריכים להתקיים על הסביבה\\mdp על מנת שתמיד נצליח למצוא את המדיניות האופטימלית.

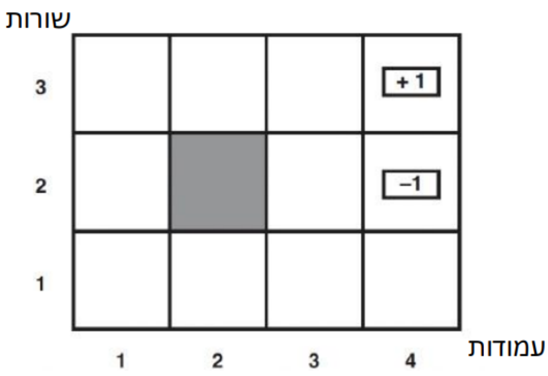
function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} [\, R(s,a) + \gamma \sum_{s'} P(s'|s,a) * U(s')] > R(s,\pi(s)) + \gamma \sum_{s'} P(s'|s,\pi(s)) * U(s')$ **do**
        $\pi[s] \leftarrow \max_{a \in A(s)} [\, R(s,a) + \gamma \sum_{s'} P(s'|s,a) * U(s')]$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

אם $\gamma = 1$: אין צורך לשנות את האלגוריתם, ההתכנסות מובטחת בpolicy_iteration אם מרחב המצבים סופי, מספר הפעולות סופי, פונקציית התגמולים חסומה, קיימים מצבים סופיים ואין מעגל תגמולים חיובי.
במקרה זה גם אם $\gamma = 1$ מובטחת התכנסות למדיניות אופטימלית (כיוון שמרחב המדיוניות סופי) גם ללא שינוי באלגוריתם.

שאלה 2:

נתון ה־MDP הבא $<\mathbf{S},\mathbf{A},\mathbf{P},\mathbf{R},\boldsymbol{\gamma}>$, אופק אינסופי:



מצבים:

$$\mathbf{S}=\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),(\mathbf{1},\mathbf{3}),(\mathbf{1},\mathbf{4}),(\mathbf{2},\mathbf{1}),(\mathbf{2},\mathbf{3}),(\mathbf{2},\mathbf{4}),(\mathbf{3},\mathbf{1}),(\mathbf{3},\mathbf{2}),(\mathbf{3},\mathbf{3}),(\mathbf{3},\mathbf{4})\}$$

$$\mathbf{S_G}=\{(\mathbf{2},\mathbf{4}),(\mathbf{3},\mathbf{4})\}$$

פעולות

$$\forall \mathbf{S}\backslash\mathbf{S_G}:\mathbf{A}(\boldsymbol{s})=\{\mathbf{Up},\mathbf{Down},\mathbf{Left},\mathbf{Right}\}$$

תגמולים:

נתונים התגמולים של המצבים הסופיים בלבד: $\mathbf{R}\big((\mathbf{2},\mathbf{4})\big)=-\mathbf{1},\mathbf{R}\big((\mathbf{3},\mathbf{4})\big)=+\mathbf{1}$.

שימו לב, התגמולים הינם תגמולים על המצבים.

ישנם תגמולים עבור שאר המצבים, הם פשוט לא נתונים כחלק מהשאלה.

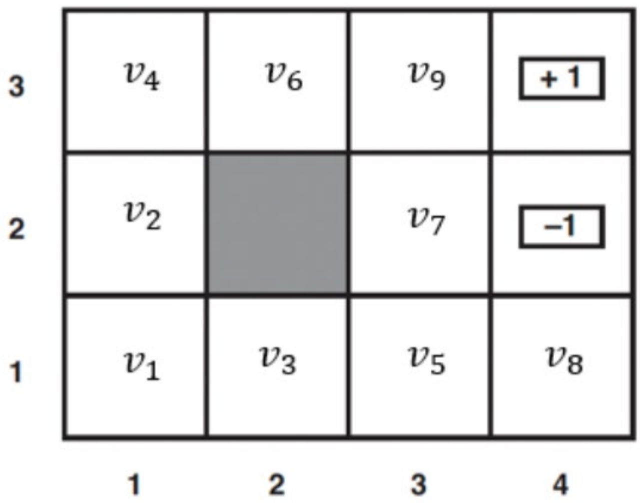
מודל מעבר:

כל פעולה "מצליחה" בהסתברות 0.8, ואם היא לא מצליחה אז בהסתברות שווה מתבצעת אחת הפעולות המאונכות לפעולה המתבקשת. כאשר הסוכן הולך לכיוון הקיר או מחוץ ללוח הוא נשאר במקום.

מקדם דעיכה: $\mathbf{0}<\boldsymbol{\gamma}<\mathbf{1}$.

הרצתם את האלגוריתם **value iteration** עם $\varepsilon \to 0$ וקיבלתם את הפלט הבא:

(משמעות הדבר ש־$\varepsilon \to 0$ היא שתנאי העצירה קַיֵּם שנורמה האינסוף בין ווקטורי התועלת הייתה אפסית, כלומר לאחר הריצה ערכי התועלת שהתקבלו מקיימים את משוואת בלמן).



כאשר v_i הוא ערך התועלת למצב ה־i כפי שניתן לראות בתרשים. בנוסף נסמן את התגמול למצב ה־i ב־r_i.

ענו נכון \ לא נכון, וספקו הסבר קצר או דוגמה נגדית מפורטת.

א. (3 נק') אם $v_9 > 1$, אז בהכרח מתקיים ש־$r_9 > 1$. נכון \ לא נכון.

דוגמה נגדית:

נגדיר $\gamma = 0.9$ ונביט בלוח התגמולים:

-0.04	-0.04	0.7	+1
-0.04	WALL	0.9	-1
-0.04	-0.04	-0.04	-0.04

נריץ את אלגוריתם value_iteration עם אפסילון ששואף ל-0 ונקבל לוח התועלות:

4.881	5.695	6.542	1
4.237	WALL	7.277	-1
4.573	5.305	6.098	4.682

והמדיניות:

RIGHT	RIGHT	DOWN	1
UP	WALL	LEFT	-1
RIGHT	RIGHT	UP	LEFT

נראה כי התועלות שמתקבלות מקיימות את משוואת בלמן:

$$v1 = -0.04 + 0.9 * \max(\ 0.8v2 + 0.1v1 + 0.1v3, \quad 0.8v1 + 0.1v1 + 0.1v3, \quad 0.8v3 + 0.1v1 + 0.1v2, \quad 0.8v1 + 0.1v1 + 0.1v2\ ) = 4.573$$

$$v2 = -0.04 + 0.9 * \max(\ 0.8v4 + 0.1v2 + 0.1v2, \quad 0.8v1 + 0.1v2 + 0.1v2, \quad 0.8v2 + 0.1v1 + 0.1v4, \quad 0.8v2 + 0.1v1 + 0.1v4\ ) = 4.237$$

$$v3 = -0.04 + 0.9 * \max(\ 0.8v3 + 0.1v1 + 0.1v5, \quad 0.8v3 + 0.1v1 + 0.1v5, \quad 0.8v1 + 0.1v3 + 0.1v3, \quad 0.8v5 + 0.1v3 + 0.1v3\ ) = 5.305$$

$$v4 = -0.04 + 0.9 * \max(\ 0.8v4 + 0.1v4 + 0.1v6, \quad 0.8v2 + 0.1v4 + 0.1v6, \quad 0.8v4 + 0.1v4 + 0.1v2, \quad 0.8v6 + 0.1v4 + 0.1v2\ ) = 4.881$$

$$v5 = -0.04 + 0.9 * \max(\ 0.8v7 + 0.1v3 + 0.1v8, \quad 0.8v5 + 0.1v3 + 0.1v8, \quad 0.8v3 + 0.1v5 + 0.1v7, \quad 0.8v8 + 0.1v5 + 0.1v7\ ) = 6.098$$

$$v6 = -0.04 + 0.9 * \max(\ 0.8v6 + 0.1v4 + 0.1v9, \quad 0.8v6 + 0.1v4 + 0.1v9, \quad 0.8v4 + 0.1v6 + 0.1v6, \quad 0.8v9 + 0.1v6 + 0.1v6\ ) = 5.695$$

$$v7 = 0.9 + 0.9 * \max(\ 0.8v9 + 0.1 * -1 + 0.1v7, \quad 0.8v5 + 0.1v7 + 0.1 * -1, \quad 0.8v7 + 0.1v9 + 0.1v5, \quad 0.8 * -1 + 0.1v9 + 0.1v5\ ) = 7.277$$

$$v9 = 0.7 + 0.9 * \max(\ 0.8v9 + 0.1 * 1 + 0.1v6, \quad 0.8v7 + 0.1v6 + 0.1 * 1, \quad 0.8v6 + 0.1v9 + 0.1v7, \quad 0.8 * 1 + 0.1v9 + 0.1v7\ ) = 6.542$$

מתקיים כי משוואת בלמן אכן מתקיימת עבור 9 המצבים אבל r9=0.7 קטן מ1 אע"פ שv9=6.542 שזה גדול מ1.

ב. (3 נק') אם $\mathbf{\forall i \in [9]: v_i > 0}$, אז בהכרח $\mathbf{\exists i \in [9]: r_i > 0}$. נכון \ לא נכון.

דוגמה נגדית:

לא נכון, עבור אותה דוגמה מהסעיף הקודם נראה כי $\forall v_i \in [1,9]: v_i > 0$ אבל לא מתקיים שכל התגמולים חיוביים.

ג. (3 נק') אם $\mathbf{r_1 = r_2 = \cdots = r_9 < 0}$, אז בהכרח $\mathbf{v_1 = \min\{v_i | i \in [9]\}}$. נכון \ לא נכון.

דוגמה נגדית:

נשתמש באותה דוגמה שראינו בתרגול:

טבלת התגמולים:

-0.04	-0.04	-0.04	1
-0.04	WALL	-0.04	-1
-0.04	-0.04	-0.04	-0.04

עבורה קיבלנו טבלת התועלות:

0.812	0.868	0.918	1
0.762	WALL	0.660	-1
0.705	0.655	0.611	0.388

קיבלנו כי $\min\{v_i \mid i \in [1,9]\} = 0.388$ אבל מתקיים כי $v_1 = 0.705 > 0.388$ לכן הטענה לא נכונה.

ד. (3 נק') אם $\mathbf{v_1 > v_2 > v_3 > 0}$, אז בהכרח $\mathbf{Up} = \boldsymbol{\pi}^*\big((\mathbf{1},\mathbf{1})\big)$. נכון \ לא נכון.

דוגמה נגדית:

ניקח את הערכים הבאים: $v1 = 3, \quad v2 = 2, \quad v3 = 1$, נראה כי בבדי שמשוואת בלמן תתקיים צריך לעשות את הפעולה LEFT :

$$v1 = r1 + \gamma * \max(\ 0.8v2 + 0.1v1 + 0.1v3, \quad 0.8v1 + 0.1v1 + 0.1v3,$$
$$0.8v3 + 0.1v1 + 0.1v2, \quad 0.8v1 + 0.1v1 + 0.1v2\ )$$

כלומר הפעולה שתיבחר במדיניות האופטימלית היא זאת שמקיימת:

$$\max(\ 0.8v2 + 0.1v1 + 0.1v3, \quad 0.8v1 + 0.1v1 + 0.1v3, \quad 0.8v3 + 0.1v1 + 0.1v2,$$
$$0.8v1 + 0.1v1 + 0.1v2\ )$$

נמצא איזה פעולה מהווה את המקסימים:

$$up \rightarrow\ 0.8v2 + 0.1v1 + 0.1v3 = 0.8 * 2 + 0.1 * 3 + 0.1 * 1 = 2$$

$$down \rightarrow \ 0.8v1 + 0.1v1 + 0.1v3 = 2.8$$

$$left \rightarrow \ \ 0.8v1 + 0.1v1 + 0.1v2 = 2.9$$

$$right \rightarrow 0.8v3 + 0.1v1 + 0.1v2 = 1.3$$

כלומר הפעולה שממקסמת את v1 היא LEFT ולא UP.

ה. (2 נק') אם $\gamma = 0$, מה מספר המדיניות האופטימליות הקיימות? נמקו.

אם $\gamma = 0$ זאת אומרת שהביטוי שנקבל עבור התועלת הוא:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) * U(s') = R(s)$$

כלומר עבור כל מצב לא משנה לנו מהי התועלת העתידית אלא רק מה שמרוויחים מיידית, ולכן בהתאם לתועלת הבאה המקסימלית נבחר הפעולה הבאה, ישנם 12 מצב בלוח שלנו, 2 מהם מצבי מטרה ואחד קיר שאי אפשר לגשת אליו, נשארים 9 מצבים.

מכל מצב ניתן לבצע 4 פעולות (למעלה, למטה, ימינה או שמאלה) ולפי הimmediate reward של כל אחד מהמצבים העוקבים לאחר הפעלת הפעולות הנ"ל, נקבל את הפעולה האופטימלית, כלומר אם:

immediate reward of next state if we choose to go up > immediate reward of next state if we choose to go left

אז נעדיף פעולת up על left, והפעולה האופטימלית תהיה זאת בעלת הreward המקסימלית, ולכן מכל מצב אפשר לעשות 4 פעולות שכל אחת מהן יכולה להיות אופטימלית בהתאם לreward שלה.

סה"כ נקבל כי מספר המדיניות האופטימליות הוא 9^4.

ו. (2 נק') לסעיף זה בלבד נתון כי $\mathbf{r_8 = 0}$, $\mathbf{v_5 = -1}$. מהו $\boldsymbol{\pi^*}\big((\mathbf{1,4})\big)$? ציינו את כל האפשרויות ונמקו.

$$\pi(s) = \underset{a\in A(s)}{\mathrm{argmax}} \sum_{s'} P(s'|s,a) * U(s')$$

נמצא עבור כל אחד מהפעולות $a \in A(s)$ את $\sum_{s'} P(s'|s,a) * U(s')$ = a_{ut}:

$$a = up \rightarrow up_{ut} = 0.8 * -1 + 0.1 * v5 + 0.1 * v8 = -0.9 + 0.1v8$$

$$a = down \rightarrow down_{ut} = 0.8 * v8 + 0.1 * v5 + 0.1 * v8 = -0.1 + 0.9v8$$

$$a = left \rightarrow left_{ut} = 0.8 * v5 + 0.1 * -1 + 0.1 * v8 = -0.9 + 0.1v8$$

$$a = right \rightarrow right_{ut} = 0.8 * v8 + 0.1 * -1 + 0.1 * v8 = -0.1 + 0.9v8$$

$$\rightarrow \quad up_{ut} = left_{ut} \quad , \quad down_{ut} = right_{ut}$$

עבור v8 > -1 הצעד האופטימלי יהיה ללכת ימינה או למטה, אחרת הצעד האופטימלי יהיה ללכת שמאלה או למעלה.

נראה עכשיו לפי משוואת בלמן כי אי אפשר שיתקיים v8< -1:

$$\begin{aligned} v8 &= r8 + \gamma \max_{a\in A(s)} \sum_{s'} P(s'|s8,a) * U(s') \\ &= r8 + \gamma * \max(-0.9 + 0.1v8, \quad -0.1 + 0.9v8) \\ &\geq r8 + \gamma * (-0.9 + 0.1v8) = 0 - 0.9\gamma + 0.1\gamma v8 \end{aligned}$$

עבור v8 <= -1 נקבל:

$$v8 \geq -0.9\gamma + 0.1\gamma v8 \geq -0.9\gamma - 0.1\gamma$$

$$v8 \leq -1 \rightarrow -1 \geq -0.9\gamma - 0.1\gamma \rightarrow \gamma \geq 1$$

בסתירה לנתון ש-$\gamma \in (0,1)$, ולכן נקבל כי הפעולות האופטימליות הן רק down, right עבור מצב (1,4).

ז. (2 נק') נתון כי $\mathbf{v_1} > \mathbf{v_2} > \mathbf{v_3} > \mathbf{0}$, מצאו חסמים צמודים, עליון ותחתון ל־$\mathbf{r_1}$ כפונקציה של $\mathbf{v_i}$ (ולא כפונקציה של $\boldsymbol{\gamma}$).

$$v1 = r1 + \gamma * \max(\ 0.8v2 + 0.1v1 + 0.1v3, \quad 0.8v1 + 0.1v1 + 0.1v3, \quad 0.8v3 + 0.1v1 + 0.1v2, \quad 0.8v1 + 0.1v1 + 0.1v2\ )$$

נתון כי $v1 > v2 > v3$ לכן :

$$\max(\ 0.8v2 + 0.1v1 + 0.1v3, \quad 0.8v1 + 0.1v1 + 0.1v3, \quad 0.8v3 + 0.1v1 + 0.1v2, \quad 0.8v1 + 0.1v1 + 0.1v2\ ) = 0.9v1 + 0.1v2$$

ולכן נקבל:

$$v1 = r1 + \gamma(0.9v1 + 0.1v2)$$

$$r1 = v1 - \gamma(0.9v1 + 0.1v2)$$

נתון כי $\gamma \in (0,1)$ ולכן נקבל:

$$r1_{max} = v1 - \gamma_{min}(0.9v1 + 0.1v2) < v1 - 0 = v1$$

$$r1_{min} = v1 - \gamma_{max}(0.9v1 + 0.1v2) > v1 - 1 * (0.9v1 + 0.1v2) = 0.1(v1 - v2)$$

כלומר:

$$0.1(v1 - v2) < r1 < v1$$

חלק למידה

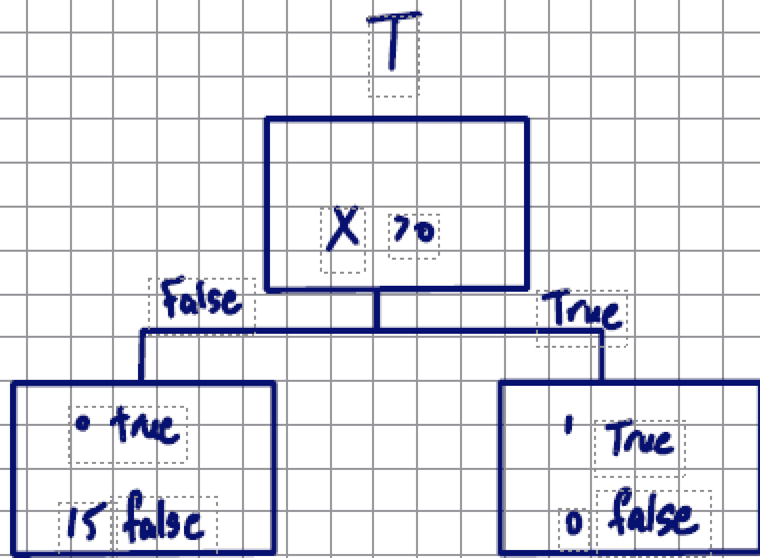
א. ניקח את הנקודות הבאות : נקודות האימון {נק1:(9,1)+, נק 2 (7,5)-} (K=1,d=2)
דוגמת מבחן : (1,1)
כך שבמרחק אוקלידי נקודה 2 היא הקרובה ולפיכך התוצאה תהיה - , ובמרחק מנהטן נקודה 1 היא הקרובה כלומר התוצאה תהיה +

ב. אם נבחר K=1 נקבל ששגיאת האימון תהיה 0 כי הנקודה הנוכחית תהיה הקרובה ביותר וכך שכל נקודה אחרת המרחק המינימלי מהנקודה הנוכחית יהיה SQRT(2)
ג. אם נבחר K=14 (כלומר במספר נקודות סט האימון) אנחנו תמיד ניקח את הסיווג הנפוץ ביותר מקבוצת האימון
ד. כפי שנלמד בתרגול: אם K יהיה קטן מדי יהיה לנו מאוד קשה להתמודד עם רעש(overfitting) ואם K גדול מדי אנחנו נתחשב גם בשכנים שלא באמת קרובים
ה. הפרכה :
נניח שיש לנו N נקודות של סט אימון, יהי d and k כלשהם ונניח נקודת מבחן((x_l,y_l) כך ש x_l שייך למרחב R^D
ונניח שאין רדיוס שייביא את אותו הסיווג כמו באלגו המקורי , כלומר בגרסה החדשה האלגו לא יצליח להביא את הסיווג של האלגוריתם המקורי, כלומר מספר השכנים שלנו יביא שיהיה לנו תיקו בין שתי התוצאות, ולפי נתוני השאלה, דבר זה מחייב שהסיווג יהיה חיובי, לפי נתוני השאלה כך שאין שתי נקודות בעלות אותו מרחק מנקודה המבחן אז יהיה לנו רדיוס מסויים כך שנפריד בין הנקודה ה-K לנקודה שאחריה, ולפיכך סתירה !
ו. הפרכה:
נניח שיש לנו N נקודות של סט אימון, יהי d and r כלשהם ונניח נקודת מבחן((x_l,y_l) כך ש x_l שייך למרחב R^D
נניח שאין k שיניב את אותה תוצאה כמו האלגוריתם החדש כלומר אין מספר שכנים קבוע שיניב את אותה תוצאה כמו האלגוריתם החדש, אבל עם הנחת השאלה נקבל סתירה מידיית, כי אם אין שתי נקודות בעלות אותו מרחק ויש רדיוס קבוע שמניב תוצאה מסויימת אז בהכרח יש מספר מסויים וקבוע של שכנים שגם יניבו את אותה תוצאה מקורית(בגלל שבאלגוריתם החדש אנחנו לוקחים גם את הקרובים ביותר על הרדיוס הנון, אז אותם שכנים באלגו המקורי יילקחו)

<u>מתפצלים ונהנים :</u>

(epsilon’=-10epsilon)(d=1)אם ניקח את ערך האפסילון הנל מתקיים שהתנאי אף פעם לא מתקיים בעץ T כך שהעץ עם ערך x תמיד יונב כ true בעץ T , אבל בעץ T’ תמיד נקבל ערך false , סתירה

E. -10 E



2b

```
Test Accuracy: 94.69%
```

3a המטרה היא להחליש את התאמת היתר על ידי ולעשות עצים לא עקביים במטרה להתמודד עם הרעש של הנתונים

3d כן שיפר

```
Test Accuracy: 97.35%
```