**חלק א'**

בחירת מעבדה אחרונה

סדר מעבר בדירות- כל סדר אפשרי בין דירות

בחירת 0 או 1 מעבדות לפני כל דירה-

m אפשרויות לכל מעבדה ועוד 1 לאף מעבדה.

1. ---

|  |  |  |  |
| --- | --- | --- | --- |
| k | m | #possiblePaths | Estimated calculation time |
| 7 | 2 | ~22.04\*10^6 | ~18.5 [secs] |
| 7 | 3 | ~24.77\*10^7 | ~3.8 [mins] |
| 8 | 3 | ~79.27\*10^8 | ~2.3 [hours] |
| 8 | 4 | ~63.00\*10^9 | ~19.6 [hours] |
| 9 | 3 | ~28.53\*10^10 | ~3.7 [hours] |
| 10 | 3 | ~11.41\*10^12 | ~5.3 [months] |
| 11 | 3 | ~50.22\*10^13 | ~20.8 [years] |
| 12 | 3 | ~24.11\*10^15 | ~1.1 [thousand years] |
| 12 | 4 | ~46.78\*10^16 | ~22.1 [thousand years] |
| 13 | 4 | ~30.41\*10^18 | ~1.5 [million years] |

**חלק ג'**

1. דרגת יציאת מינימלית – 0:  
   ממצב מטרה נעבור בכל המעבדות שעדיין לא עברנו בהן, כעת לא ניתן לבקר באף דירה/ מעבדה.

דרגת יציאה מקסימלית – k+m:

מהמצב ההתחלתי, בהנחה שיש מספיק מטושים, ניתן לעבור לכל אחת מהדירות/ מעבדות.

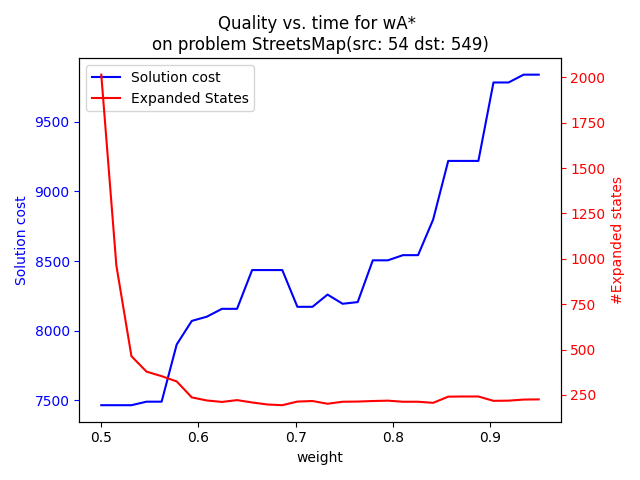
1. לא ייתכנו במעגלים, נניח בשלילה שיש מעגל ונחלק למקרים:  
   - נניח שעל המעגל יש ביקור במעבדה שלא היינו בה, הביקור בה יוביל להוספה שלה ל – VisitedLabs -> כעת לא ניתן לחזור במעגל למצב שבו לא ביקרנו במעבדה, וזו סתירה.  
   - נניח שעל המעגל יש ביקור במעבדה שהיינו בה -> חייבות להיות בדיקות(דירות) ב – Taken שהן לא ב Transferred כדי לבקר במעבדה, אחרי הביקור במעבדה הבדיקות יתווספו ל Transferred -> כעת לא ניתן לחזור למצב שבו הבדיקות הנ"ל לא נמצאות ב Transferred, וזו סתירה.  
   - נניח שעל המעגל יש ביקור דירה ולפי ההגדרה היא לא ב - Taken, הדירה תתווסף ל Taken ולא ניתן לחזור למצב שהיא לא נמצאת בו, כי לא ניתן לעבור במעבדה וע"י כך להוריד אותה מ – Taken כי כפי שראינו אין מעבדות על המעגל.
2. יש במרחב אינסוף מצבים, זאת מכיוון שאין חסם מוגדר על כמות המטושים, למשל אינסוף מצבים שזהים למצב ההתחלתי אך נבדלים ממנו רק במספר המטושים. כל מצב שכזה שבו מספר המטושים גדול ממספר המטושים של המצב ההתחלתי + כל המטושים שבמעבדות הוא בהכרח לא ישיג.
3. – האם יש מספיק מטושים לבדוק את כולם? (כן ייתכן, אם סכום כל המטושים קטן ממס' השותפים בכל הדירות ביקרנו, אז גם אם ביקרנו בכל המעבדות, לא ניתן לעבור בכל הדירות כי אין לנו מספיק מטושים).
4. אורך מינימלי של מסלול במרחב ממצב התחלתי לסופי : k+1.  
   נניח שיש מספיק מטושים עבור כל השותפים שבדירות הקיימות, ושיש מספיק מקום במקררים באמבולנס, אז ניתן ע"י מעבר מהמצב ההתחלתי לכל הדירות ולבסוף להעביר למעבדה ולהגיע למצב סופי.

אורך מקסימלי של מסלול במרחב ממצב התחלתי לסופי : .  
תחילה מהמצב ההתחלתי נעבור בכל מעבדה פעם אחת לפני הביקור בדירות (איסוף מטושים), - m מעברים, ואחרי כל מעבר בדירה נוריד את הבדיקה במעבדה *– 2k מעברים.*

***חלק ה'***

1. *מס' הפיתוחים בריצה העיוורת – 17354*

*מס' הפיתוחים בריצה מיודעת (יוריסטיקת מרחק אווירי) – 2015.  
 לכן, חסכנו 15339 פיתוחים, שזה 88.4% (15339/17354)*

**

*כפי שניתן לראות, הפתרון הטוב ביותר מתקבל עבור w=0.5, כפי שלמדנו בהרצאה שזה Astar רגיל והוא אלגוריתם קביל עבור יוריסטיקה קבילה (שהיא אכן כך במקרה הנ"ל).  
המגמה הכללית אומרת שככל שהמשקל עולה איכות הפתרון יורדת. בגרף ניתן לראות שזאת המגמה, אך היא לא תמיד נכונה, למשל עבור weight=0.67 מחיר הפתרון הוא 8500 אבל עבור weight=0.7 מחיר הפתרון הוא ~8250.  
בנוסף, המגמה הכללית גם אומרת שככל שהמשקל עולה מס' הפיתוחים קטן. שוב, בגרף ניתן לראות שזאת המגמה, אך היא לא תמיד נכונה, למשל עבור weight=0.84, מס' הפיתוחים שבוצעו הוא 200 אבל עבור weight=0.86 מס' הפיתוחים שבוצעו הוא 230.*

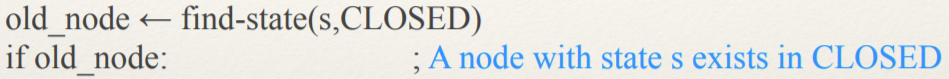
***חלק ו'***

*19. בשימוש בתת הבעיה השתמשנו ב – caching עבור פיתוח מצבים במרחב החיפוש של המפה, דבר שלא יתאפשר במרחב החיפוש של ה – MDA. לכן, פיתוח כל מצב כמתואר בגישה הזאת ייקח יותר זמן (חישוב המרחקים מחדש).*

*20. שורת הקוד הרלוונטית:*

@dataclass(frozen=True)

*שורה זו לא מספיקה, משום שניתן יהיה עדיין לשנות את השדות של האובייקט, ההגנות הנוספות שיש בקוד הוא שימוש לדוגמא ב – FrozenSet עבור Set שהוא מבנה נתונים שניתן לגשת אליו by reference. (וה frozen מבטל אפשרות לשינויים כנדרש).  
  
ייתכן באלג' A\* שנפגוש מצב בשנית גם לאחר שפיתחנו אותו, להלן השורה מהאלגוריתם מההרצאה:*



אנו רוצים לעשות זאת עבור הטיפוס MDAState כי למשל, כל פעם שאנחנו חוקרים את המצב הבא לפיתוח אנחנו מעבירים לו set שהוא איחוד של test\_on\_ambulance עם הדירה הנוכחית שנבדקה, אם היינו משתמשים ב set במקום frozenset כל השינויים היו עלולים לשנות את ה set המקורי, ואז אם היינו רוצים לפתח מחדש מצב שכבר פותח, לא ניתן יהיה לשחזר את ה set המתאים.  
הבאג יכול לקרות ע"י מימוש כנ"ל לדוגמא:

tests\_on\_ambulance = state\_to\_expand.tests\_on\_ambulance  
tests\_on\_ambulance = tests\_on\_ambulance.union(set([apartment]))

23. נוכיח כי בכל מצב S משקל המסלול האופטימלי מ-S למצב סופי (נסמנו p) קטן או שווה לערך ההיוריסטיקה MaxAirDist, נסמנו M.

נניח מצב S בו אנו נמצאים בצומת j (או מעבדה או דירה או צומת כללית) ולא ביקרנו בדירות d1,d­2,..,dk­ לכן המסלול האופטימלי עובר בדירות אלה משום שהוא מסלול למצב סופי לפי הגדרה זהו מצב בו ביקרנו בכל הדירות. M המרחק המקסימלי בין זוג דירות נסמנן di,dj.נניח בשלילה שמשקל p קטן מ-M בפרט di,dj שתי דירות במסלול p, כיוון וכל המשקלים חיוביים, משקל המסלול ביניהן קטן שווה למשקל p, לכן קטן מהמרחק האווירי ביניהן וזו סתירה, לכן ההיוריסטיקה אופטימית, היא גם אי שלילית ולכן קבילה.

(יש לבדוק שוב נכונות דוגמה נגדית זו בהתאם להנחיות התרגיל, יש לבדוק האם צריך טבלת מעקב בתרגיל זה).

26. נציג דוגמה נגדית בה קיים מסלול ממצב S למצב סופי בעל משקל נמוך ממשקל ההיוריסטיקה SumAirDist עבור אותו מצב, בפרט משקל מסלול אופטימלי קטן ממשקל המסלול שנראה ולכן גם הוא קטן ממשקל ההיוריסטיקה, לכן היא אינה אופטימית ולא קבילה

נגדיר מרחב כבישים בו

נגדיר את (0,0) צומת התחלה עבור בעיית MDA, (2.1,0) מעבדה וכל שאר הצמתים הם דירות.

נגדיר מספר מטושים התחלתי להיות 100, מספר השותפים בדירה הוא 1 לכל אחת מהדירות, באמבולנס מקרר אחד עם קיבולת של 100 בדיקות. כיוון ויש כביש ישיר בין כל שתי נקודות במרחב המפה, המרחק הקטן ביותר בין כל שני צמתים הוא המרחק האווירי ביניהן.

sumAirDist יחזיר את אורך המסלול בו מתחילים מצומת ההתחלה וכל פעם עוברים לדירה הקרובה ביותר בין הדירות שנותרו (עבור המצב ההתחלתי מדובר בכל הדירות בבעיה), במסלול זה פונקצית המשקל על מעבר בין שתי דירות היא מרחק אווירי ביניהן, בבעיה שהגדרנו מדובר באותו ערך כמו המרחק הקצר ביותר בין שני צמתים במרחב המפה.

לכן הערך שנקבל הוא 5, משום שהנקודה (0.5,0) היא הקרובה ביותר לצומת ההתחלה ונמצאת במרחק 0.5, הדירה הקרובה ביותר לזו נמצאת בנקודה (1,0) במרחק 0.5, הדירה הקרובה ביותר לזו נמצאת בנקודה (2,0) במרחק 1, הדירה האחרונה שנותרה (לכן גם הקרובה ביותר) נמצאת בנקודה (-1,0) במרחק 3.

נסמן את הדירות במרחב d1,…,d4 מסודרות לפי ערכי x מהקטן לגדול

נסמן את המעבדה l, במעבדה זו אין מטושים לחלוקה.

נביט במסלול הבא במרחב MDA, נתחיל מצומת ההתחלה ומשם נעבור בדירה במיקום (-1,0), משם נמשיך

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Remaining capacity | Visited Labs | #matoshim | Tests transferred | Tests Taken | Location |
| 100 | {} | 100 | {} | {} | (0,0) |
| 99 | {} | 99 | {} | {d1} | (-1,0) |
| 98 | {} | 98 | {} | {d1,d2} | (0.5,0) |
| 97 | {} | 97 | {} | {d1,d2,d3} | (1,0) |
| 96 | {} | 96 | {} | {d1,d2,d3,d4} | (2,0) |
| 100 | {l} | 96 | {d1,d2,d3,d4} | {} | (2.1,0) |

מסלול זה חוקי, משום שביקרנו במעבדה שלא היינו בה קודם, ובכל דירה שביקרנו לא היינו קודם לכן, וגם היו לנו מספיק מטושים וקיבולת מספיק גדולה לאחסן את כל הבדיקות בכל דירה שביקרנו בה.

משקל מסלול זה לפי פונקצית המרחק הוא סכום המרחקים שעברנו בכל ביקור, כאשר זה המרחק הקטן ביותר במרחב הכבישים, ושם כבר ראינו כי במקרה זה מדובר במרחק אווירי בין שתי נקודות.

לכן המשקל של המסלול שהצגנו הוא 4.1, זה קטן מאשר ערך ההיוריסטיקה לכן היוריסטיקה זו אינה קבילה.

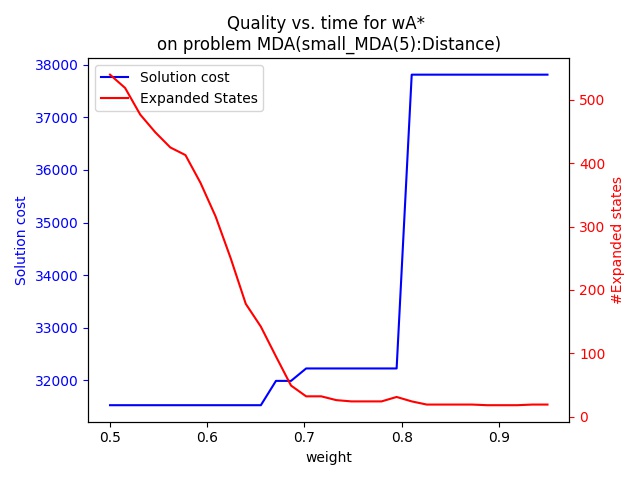
29.נוכיח כי לכל מצב S משקל המסלול האופטימלי (נסמנו p) מ-S למצב סופי קטן או שווה לערך ההיוריסטיקה MSTAirDist עבור S נסמנו M.

נניח כי במצב S המיקום הוא הצומת j והדירות שנותר לבקר בהן הן d1,d2,…,dk, מכיוון ש-p מסלול למצב סופי, הדירות הללו מופיעות בו.  
נניח בשלילה שמשקל מסלול (סכום משקל כל קשתות) p קטן ממש מ-M (משקל עפ"מ עבור גרף **הצמתים** Gj,d1,..,dk כאשר משקל כל קשת היא מרחק אווירי בין שני צמתים וקיימת קשת בין כל שני צמתים). נגדיר: A={j=d0,d1,…,dk}

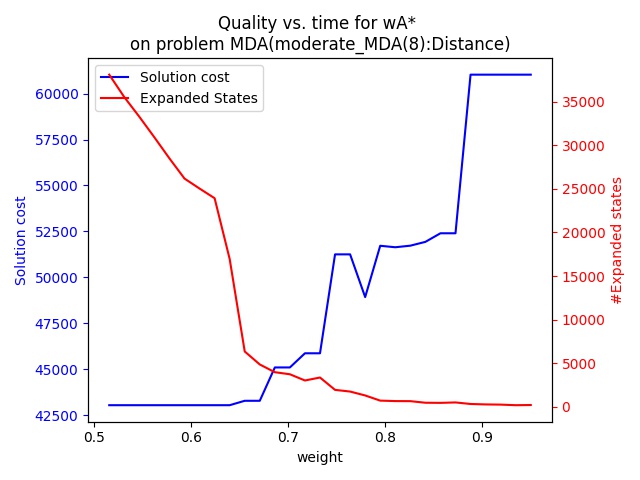
יהיו di->…->dj תת מסלול של p העובר שתי צמתים סמוכים ב-A (ביניהן אין מעבר באף צומת אחר ב-A במסלול), משקל המסלול יהיה גדול או שווה למרחק האווירי ביניהן.  
נגדיר p’ מסלול המושרה על ידי p כך שלכל שתי צמתים ב- A , diו- dj­ כך שקיים מסלול ב-p מ-di ל- dj ובו אין אף צומת אחרת ב-A, תהיה קשת ישירה ב- p’ בין di ל-dj.  
המסלול p’ הוא עץ פורש של G משום שכל צמתיו ב-G, כל צמתי G נמצאים בו ואין בו מעגלים, משקל מסלול p’ כאשר משקל כל קשת בו הוא מרחק אווירי בין שני צמתים בו, קטן או שווה למשקל מסלול p משום שלכל קשת di->dj ב-p’ קיים מסלול di->…->dj ב-p שמשקלו גדול או שווה למשקל הקשת di->dj, לכן אם משקל p קטן מ-M אזי גם משקל p’ קטן מ-M, לכן מצאנו עץ פורש עם משקל נמוך יותר מעפ"מ עבור הגרף G וזו סתירה.

לכן לכל צומת בגרף, ההיוריסטיקה אופטימית, היא גם אי שלילית ולכן קבילה.

30.



בגרף רואים כי עבור משקלים קטנים מקבלים פתרון אופטימלי או כמעט אופטימלי אבל במחיר פיתוח רב של מצבים, עבור משקלים גדולים (קרובים ל-1) מפתחים מעט מצבים אבל משלמים במחיר הפתרון, עבור פתרונות בתחום 0.7-0.8 מקבלים פתרון שקרוב לפתרון האופטימלי ומפתחים מעט מאוד מצבים, זהו תחום משקלים כדאי.



בגרף זה רואים שמקבלים פתרון טוב במחיר פיתוח מעט צמתים בתחום המשקלים 0.67-0.73 זהו תחום כדאי יותר משאר האיזורים בהם או שהפתרון אינו טוב דיו או שמפתחים מספר רב מאוד של צמתים כדי למצוא אותו.

**חלק ז'**

31.

|  |  |  |  |
| --- | --- | --- | --- |
|  | MDAMaxAirDistHeuristic | MDASumAirDistHeuristic | MDAMSTAirDistHeuristic |
| CostMDAtest travel | לא | לא | לא |
| CostMDAmonetary | לא | לא | לא |

32.

עבור פתרון בעיית מד"א הקטנה בחיפוש UnifromCost עבור מחיר המרחק, קיבלנו שמחיר המוניטרי היה 49.7₪ ועבור בעיית אופטימיזציה למחיר מוניטרי קיבלנו מחיר של 42.5₪.

עבור הבעיה הבינונית אופטימיזציה למרחק קיבלנו מחיר של 95.847₪ ועבור אופטימיזציה למחיר מוניטרי קיבלנו מחיר של 77.20101₪.

34.

נביט במסלול אופטימלי ממצב S למצב סופי, במצב S אנו מתחילים בצומת j=d0  עם מספר מטושים n0 ונותר לנו לעבור בדירות d1,…,dk עבור הדירה ה-i נסמן את מספר האנשים שיש לבדוק בה ni, ונסמן את המרחק מ-di למעבדה הבאה במסלול אחרי di, disti משקל מרחק הבדיקות הכולל עבור המסלול האופטימלי p: d0->...->d1->... ...->dk הוא .

נסמן את המרחק האווירי מ-di למעבדה הבאה אחרי di במסלול בתור AirDist*i* ידוע לנו משום שאנו במרחב האוקלידי כי .

ההיוריסטיקה מחשבת לכל צומת d*i* את המרחק האווירי הקטן ביותר של מעבדה אליו AirDist’i לכן .

סה"כ ערך ההיורסיטקיה הוא .

אנו רואים כי מתקיים לכל i, ולכן:

לכן ערך ההיוריסטיקה מכל צומת S קטן שווה למשקל מסלול אופטימלי מ-S למצב סופי   
 לכן ההיוריסטיקה אופטימית, היא גם אי שלילית ולכן קבילה.

35. עבור ההרצה כשבעיית האופטימיזציה הייתה המרחק של האמבולנס, הבדיקות עברו מרחק של 176 ק"מ, בעוד שכאשר חיפשנו פתרון עבור בעיית אופטימיזציה למרחק שעוברות הבדיקות, אנו מקבלים שהמרחק הוא 131 ק"מ.

36. יהיה מסלול במרחב המסלולים (S1)->(S1,S2)->....(S1,S2,....St), מסלול זה הוא היחיד היפתח את המצב (S1,S2,....St), נניח שקיים מסלול אחר, בו במקום (Si,Si+1) מופיעה קשת (Si, Sj), אזי לאחר Si במצב האחרון של המסלול יופיע Sj, במצב (S1,S2,....St) מופיע Si+1, לכן אכן קיים מסלול אחד (S1)->(S1,S2)->....(S1,S2,....St) שמפתח את (S1,S2,....St).

לכל מסלול במרחב מד"א p: S1->S2->....->Sk, שמקיים שמחירו C קטן מ- (1+ε)\*C\*dist, או שיימצא מסלול סופי אחר לפניו, או שהמסלול (S1)->(S1,S2)->....(S1,S2,....Sk) יימצא במרחב המסלולים, לכל רישא של p, S1->...->St משקל הרישא מבחינת מרחק כולל קטן או שווה למשקל המסלול כולו C, לכן גם משקל הרישא במרחב המסלולים מבחינת מרחק כולל קטן או שווה ל-C, לכן קטן מ-(1+ε)\*C\*dist, בנוסף הצומת שלו

(S1,S2,....St) לעולם אינה ב-close משום שהיחיד שמפתח מצב זה הוא המסלול   
(S1)->(S1,S2)->....(S1,S2,....St), לכן כל פעולת פיתוח של המסלול היא חוקית, מרחב המסלולים הוא סופי משום שמרחב המצבים הוא סופי (ראינו זאת בחלק א'), לכן או שיוחזר פתרון אחר או שיוחזר המסלול (S1)->(S1,S2)->....(S1,S2,....Sk).

לכן בפרט, אם קיים פתרון לבעיית מד"א, אז קיים פתרון אופטימלי שמשקלו C\*dist בפרט, הוא קטן מ-(1+ε)\*C\*dist, לכן או שיוחזר פתרון אחר או שפתרון זה יוחזר ובכלל, מובטח שיוחזר פתרון.

37. יהיה מסלול p: (S1)->(S1,S2)->....(S1,S2,....Sk) שמוחזר על ידי האלגוריתם, משקלו מבחינת מרחק כולל C קטן או שווה ל-(1+ε)\*C\*dist, נניח בשלילה שמשקלו גדול יותר, נסתכל על הצעד האחרון של האלגוריתם הוא פיתוח הצומת (S1,S2,....Sk), בנסיון פיתוח זה נראה כי משקל המרחק הכולל בפיתוח צומת זה חורג מהמותר ולכן הצומת לא יפותח, זו סתירה לכך שהמסלול אכן מוחזר, לכן הצומת אכן מפותח. נניח שקיים פתרון נוסף מסלול במרחב מד"א p’: S1->S’2->....->S’m שמשקלו מבחינת מרחק כולל C’ קטן או שווה ל-(1+ε)\*C\*dist, לכן לפי הטענה שהוכחנו בסעיף הקודם, קיים פתרון חוקי במרחב המסלולים p\*: (S1)->(S1,S’2)->....(S1,S’2,....S’m) נניח שמשקל מסלול זה מבחינת test-travel קטן יותר ממשקל המסלול p. נביט במצב האלגוריתם לפני ביקור במצב (S1,S2,....Sk), במצב זה עוד לא פיתחנו את המצב (S1,S’2,....S’m) משום שהוא סופי ואם נפתח אותו האלגוריתם יסתיים, לכן רישא כלשהו של המסלול (S1)->(S1,S’2)->....(S1,S’2,....S’t) פותחה והצומת (S1,S’2,....S’t+1) נמצא בתור open, כיוון ומדובר ברישא של המסלול p\*, אזי משקלו מבחינת test-travel קטן או שווה למשקל המסלול p\* כולו שקטן ממשקל המסלול p שהוא משקל הפיתוח של הצומת האחרונה שלו, לכן הצומת (S1,S’2,....S’t+1) יפותח קודם, לכן כל המסלול p\* יפותח לפני שהמצב האחרון של p מפותח, משום ש-p\* מסלול למצב סופי אז הוא יוחזר ולא המסלול p, זו סתירה לכן המסלול p הוא מסלול אופטימלי מבחינת test-travel ועוד בדרישת ההגבלה על מרחק כולל, לכן הוא אופטימלי במדד המשולב לפי הגדרת המדד המשולב.

**חלק י'**

1. זיכרון. האלגוריתם IDA\* מבצע חיפוש לעומק, מה שגורם לצריכת זיכרון לינארית באורך המסלול, בניגוד ל A\* שבו צריכת הזיכרון פרופרציונית למס' הצמתים שנוצרו בשל שמירת חזית החיפוש.
2. צמתים שפותחו

ii+iii. גם ב IDA\* וגם ב ID-DFS מדד זה נפגע משתי סיבות: חזרה על החיפוש עבור עומקים הולכים וגדלים, ומשום שלא שומרים צמתים שביקרנו בהם ב OPEN / CLOSED, ולכן הם מפתחים צמתים בהם כבר ביקרו.   
משום שב- A\* ייתכן שנפתח צמתים שהיו ב- close, ולכן צומת יכול להיות מפותח יותר מפעם אחת, לכן, מדד זה יכול להיפגע פחות ב – IDA\* לעומת A\*, מאשר ב- ID-DFS לעומת BFS. (מצד שני, ב IDA\* ייתכן שבכל פעם נוסיף רק צומת אחד (אם f שונה בין כל צומת למשל) ויהיו המון חזרות, ייתכן גם ב – ID-DFS? (שרוך?) – נרמז מסעיף ג' שלזה הכוונה.)

* 1. האלגוריתם יבצע על לכל היותר

איטרציות בלולאה שקוראת ל DFS-f.

השנוי ב- f שצריך לבצע מהערך ההתחלתי עד לערך הפתרון

בכל פעם מקדמים את f, לפי הגדרת nextFLimit בלפחות

* 1. , מכיוון שאנו נעים בקפיצות של , או מעגלים את הערך המקורי למטה (ולכן עדיין לא מגיעים לפתרון), ניתן יהיה לפספס את הפתרון שהוא ה –f-limit שנהיה בו בסוף, בלכל היותר