**AI – HW1**

חלק א'

|  |  |
| --- | --- |
| **פרמוטציות** | **K** |
| 1 | 1 |
| 6 | 2 |
| 90 | 3 |
| 2520 | 4 |
| 113400 | 5 |
| 7484400 | 6 |
| 681080400 | 7 |
| 81729648000 | 8 |
| 1.25046E+13 | 9 |
| 2.37588E+15 | 10 |

1. הטבלה:

חלק ג'

1. מקדם הסיעוף המינימלי הינו 0 כאשר המצבים שעבורם מקדם הסיעוף הינו 0 הם מצבי מטרה שעבורם מתקיים כאשר v הינה צומת על המפה, זאת משום שכאשר אין אופרטור חוקי המאפשר מעבר למצב אחר.

מקדם הסיעוף המקסימלי הינו k והוא ייתכן כאשר כל הזמנה הינה מצומת אחר במפה וכן קיים מסלול ישיר מצומת ההתחלה לכל אחד מצמתים אלו – כלומר, האוטובוס יכול לבחור כל אחד מההזמנות לאיסוף ראשון.

1. לא יכולים להיות מעגלים במרחב החיפוש משום שלפי האופרטורים החוקיים קבוצת ההזמנות הממתינות יורדת מונוטונית וכן קבוצת ההזמנות שבוצעו עולה מונוטונית ולפי הגדרת האופרטורים לפחות אחת מהן משתנה בעקבות הפעלת אופרטור ולכן אין מצב שבו יתקיים עבור שני מצבים כך וגם וכן נמצאת אחרי על אותו מסלול.
2. G הם קבוצת המצבים כך שלא נותרו על האוטובוס הזמנות וכן כל ההזמנות ברשימה המקורית בוצעו. הגדרה פורמלית:
3. לפי הגדרת האופרטורים והנתון בשאלה לפיו אין שתי הזמנות שמתחילות ונגמרות במיקומים זהים נסיק כי בכל מעבר בין מצבים מתקיים אחד מהמקרים הבאים: או שאדם אחד עולה על האוטובוס, או שאדם אחד יורד מהאוטובוס או שאדם אחד עולה ואדם אחד יורד. לכן, ב-G יש מצב עבור כל הזמנה כך שאדם זה הינו האחרון שיורד מהאוטובוס (לא יכולים להיות שני אנשים שונים היורדים באותה תחנה) => הביטוי המתאים הינו:
4. לא יתכנו בורות שאינם מצבי מטרה משום שנניח שאנו במצב S כלשהו שאינו מצב מטרה לכן בהכרח קיים מצב S’ שונה מ-S כך שבמצב זה מורידים או מעלים לפחות נוסע אחד וכן קיים מסלול במפה בין ו- (ההנחה לגבי המסלולים במפה ניתנת ב-FAQ ).
5. פונקציית העוקב:
6. הערך המקסימלי הינו מצב זה יקרה כאשר בכל תחנה או נאסוף נוסע או נוריד נוסע כלומר מספר האנשים על האוטובוס בכל רגע נתון הוא לכל היותר 1, כלומר מספר הנוסעים על האוטובוס יהיה : 0, 1, 0, 1 וכו'.

הערך המינימלי הוא k+1 במקרה זה בכל תחנה נאסוף נוסע ונוריד נוסע ולכן מלבד במצב ההתחלה ובמצב הסופי על האוטובוס יש בדיוק נוסע 1 ולכן נבצע k מעברים בכדי לאסוף k נוסעים וכן נעשה מעבר נוסף בכדי להוריד את הנוסע האחרון.

חלק ד'

1. *הקוד לוקח הזמנה באופן ראנדומלי ומדפיס את צומת המוצא וצומת היעד של ההזמנה וכן את הקורדינטות של שני הצמתים. בנוסף, הקוד מדפיס את המרחק האווירי בין צמתים אלו.*

*הפלט:*

*load\_map\_from\_csv: 21.79sec*

*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: -1.00km*

1. *הפלט שהתקבל:*

*Junction idx: #851288*

*waiting for bus: [(23695, 33320)]*

*orders on bus: [(851288, 533396), (32056, 834603)]*

*finished orders: [(47521, 606430), (466524, 29249)]*

*Is goal? False*

*חלק ה'*

1. *הערך שהתקבל הינו :77.35 ק"מ.*

**

1. *התשובה היא C.  
   לא ניתן לקבוע באופן כללי משום שהמספר המתקבל הינו סכום כל המסלולים על כל ההזמנות בנפרד, בסכום זה אין התחשבות במסלולים חופפים (לכן זה אינו חסם תחתון) וכן אין התחשבות במרחק בין הזמנה להזמנה (למשל המרחק בין S4 ל-S3 בגרף לא מחושב – ולכן זה אינו חסם עליון).*

*חלק ו'*



|  |  |
| --- | --- |
| Solution cost (total distance in KM) | Input file |
| *135.06km* | TLV\_5 |
| *539.59km* | SDEROT\_50 |
| *565.68km* | HAIFA\_100 |
| *1329.21km* | BEER\_SHEVA\_100 |

1. *חסם תחתון : 2 צמתים אם כל כל ההזמנות יוצאות מV ומגיעות ל U .*

*מ* I *יפותח רק המצב X = וממנו יפותח רק המצב Y=*

*חסם עליון :*

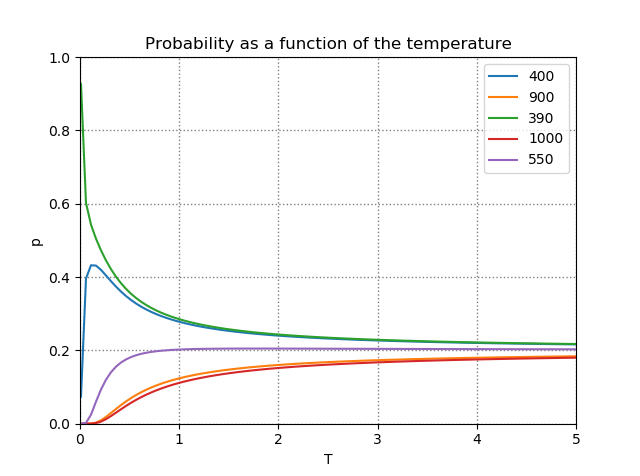
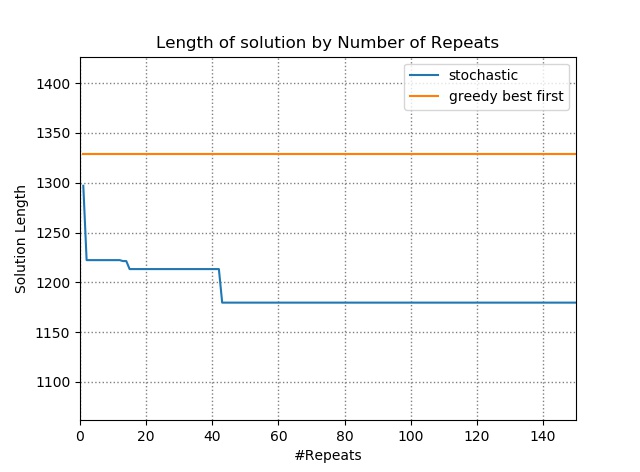
*חלק ז'*

1. *נוכיח ששינוי הסקאלה אינו משנה את התפלגות הווקטור, כלומר נראה שעבור ווקטור נתון מתקיים:*

*תהי נפריד למקרים:*

1. =>
2. =>

נשים לב כי לעולם לא יהיה 0 כי פונקציות ה-Score שלנו מחשבת מרחק אווירי ולעולם לא נקבל מרחק אווירי 0 עבור שני מצבים עוקבים – כי אז הם היו אותו מצב.

1. הגרך המתקבל:  
   
2. על פי התבוננות בגרף נראה כי כאשר נקבל כי ההסתברות של הנקודה הטובה ביותר לפי פונקציית העלות (נסמנה ) תקיים . זאת משום שכפי שצויין בחלק ז' במסמך ככל ש-T קטן יותר נבחר אפשרויות "בטוחות יותר" ובמקרה שלנו הנקודה הקרובה ביותר אווירית הינה האפשרות הבטוחה ביותר שהיא גם הקרובה ביותר בפועל.
3. על פי התבוננות בגרף נראה כי כאשר נקבל כי ההסתברות של כל אחת מ-N הנקודות הטובות יותר לפי פונקציית העלות תקיים . זאת משום שכפי שצויין בחלק ז' במסמך ככל ש-T יגדל יותר ההסתברות של כל אחת מהנקודות האפשרויות תראה לנו שווה ולכן לכל נקודה ההסתברות תשאף ל-
4. **
5. .????
6. *בdataset שלנו יש 5 הזמנות עם מוצאים ויעדים שונים כלומר 10 נקודות שונות על המפה וכולם שונות מנקודת ההתחלה של האוטובוס. מספר המסלולים עד למצב מקבל כלשהו שקול למספר הפרמוטציות החוקיות על איסוף והורדת ההזמנות כולן. בשלב ראשון ניתן לאסוף כל אחד מ5 ההזמנות. לאחר מכן ניתן לאסוף כל אחד מ4 ההזמנות האחרון או להוריד את ההזמנה שעל האוטובוס. וכן הלאה . נחשב בעזרת רקורסיה*

*נגדיר*

*numberOfRouts(n,k) =*

*כאשר n הוא מספר ההזמנות שמחכות וk מספר ההזמנות שנמצאות על האוטובוס. נוסחא זו נכונה רק עבור מצבים בהם אין חפיפה בין נקודות העלאה והורדה בהזמנות.*

*אז*

*numberOfRours(5,0) =113400*