* 1. : the number of ways to arrange the apartments between themselves
  2. : before every apartment we visit(k) we choose with repetitions which lab to visit(m) or not at all(1)
  3. : choosing the lab the ambulance visits after the last apartment
* Assuming there is a single path between every pair of points(labs and apartments)

1. Bonus

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Estimated calculation time |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

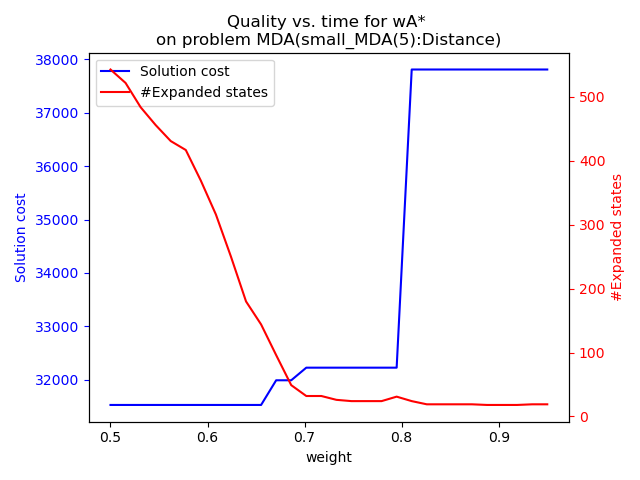
1. דרגת היציאה המינימאלית היא בקבוצת המצבים המסיימים הבאה

במילים, קבוצת המצבים הסופיים כך שבמסלול ביקרו בכל המעבדות,  
אם לא ביקרנו במעבדה מסויימת אז מהגדרת האופרטור נוכל לבקר בה, ולכן דרגת היציאה אינה 0.

דרגת יציאה מקסימלית היא והיא מתקיימת כאשר אנחנו במצב התחלתי ויש מספיק מטושים כדי לבדוק כל דירה לכן אפשר לנסוע לכל דירה() או אפשר לנסוע לכל המעבדות() לאסוף מטושים.

1. לא, נתבונן באופרטורים.  
   נניח שכן, נניח גם שקיימת קשת עם האופרטור לביקור בדירה , האופרטור מבצע ,מהגדרת האופרטור כדי לחזור ל צריך ש לא יהיה ב אך אין אופרטור שהופך את הפעולה.  
   לכן אם קיים מעגל הוא בהכרח בנוי רק מהאופרטורים , זאת אומרת שנמצאים במעבדה ולאחר כמה נסיעות(בין מעבדות) חוזרים לאותה במעבדה בלי לעבור בדירות בדרך, אך הדבר לא אפשרי מבחינת משום שלא אספנו בדיקות נוספות וגם ביקרנו במעבדה בעבר()
2. תחת ההנחה כי מספר המטושים במצב ההתחלתי הוא קבוע וכך גם מספר המטושים בכל מעבדה,
3. כן, למשל אם לאחר מעבר בכל המעבדות מספר המטושים (ההתחלתי + המטושים במעבדות) לא מספיקים לשום דירה, נתקע במעבדה האחרונה.
4. המסלול הקצר ביותר הוא באורך הוא במצב ההתחלה יש לאמבולנס מספיק מטושים כדי לבצע בדיקות לכל הדיירים בכל הדירות וגם יש באמבולנס מספיק מקום לכל הבדיקות, כך שמסלול יכלול מעבר בין כל הדירות ויסיים במעבדה.  
   המסלול הארוך ביותר הוא באורך , הוא מורכב מביקור בכל המעבדות ברצף() לאחר מכן נבקר בדירות ובמעבדות לסירוגין().

כדי ליצור את המסלול הארוך ביותר נרצה למקסם את הביקורים במעבדות, לשם כך בין כל שתי דירות נבקר במעבדה, סדרת הביקורים הזאת תביא לקשתות. ולפני זה נבקר בכל המעבדות .

1. *S*
2. *שימוש ב"מרחב על" גורם להוספה של מצבים נוספים רבים לבעיה, דבר המביא לפגיעה בביצועים משום שהוא מגדיל בצורה משמעותית את עומק הפתרונות שצריך לבדוק, בנוסף הדבר היה פוגע ביכולתנו לשמור Cache של מרחקים בין נקודות עניין(מעבדות ודירות) בגלל אורכי הפתרונות.*
   1. *@dataclass(frozen=True)*
   2. *שימוש בFrozenSet מבטיח כי לא יהיו שינויים במבני הנתונים הפנימיים*
   3. *לא, כפי שראינו בסעיף 5, לא ייתכנו מעגלים במרחב החיפוש.*
   4. *מישהו משהו*
3. *היוריסטיקה* MDAMaxAirDistHeuristic עבור אינה קבילה.
   1. *ראינו בתרגול שיוריסטיקה קבילה בהכרח מקיימת אך במקרה בו הגענו למצב מסיים על ידי מעבר בכל הדירות אך לא בכל המעבדות, כאשר נחשב את נקבל ערך שאינו 0 אלא את המרחקים במקסימלים בין המעבדות שנשארו.*
4. *היוריסטיקה* MDASumAirDistHeuristic עבור אינה קבילה.
   1. *מאותה סיבה כמו היוריסטיקה הקודמת, כאשר מחשבים נקבל מרחקים בין המעבדות שנשאר לבקר.*
5. *היוריסטיקה*MDAMSTAirDistHeuristic עבור אינה קבילה
   1. מאותה סיבה כמו היוריסטיקה הקודמת, הבעיה נובעת שחישוב המרחקים נעשה בין נקודות(צמתים) שלא צריך לבקר בהם כלל.
6. *באזור אנחנו רואים שיש אזור שבו הפתרון קרוב לאופטימלי ומספר המצבים שפותחו קרוב למינימאלי, כמובן שכפי שלמדנו אין תשובה חד משמעית. אם איכות הפתרון חשובה אז כדאי להגדיר לאומת זאת אם מהירות קבלת פתרון חשובה אז כדאי להגדיר באזור זה מתקבל פתרון שהוא לא רחוק מאופטימלי ומספר המצבים המפותח נמוך יחסית.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | MDAMaxAirDistHeuristic | MDASumAirDistHeuristic | MDAMSTAirDistHeuristic |
|  |  |  |  |
|  |  |  |  |

1. *S*
2. *השינוי בפונקציית העלות הביא לבחירת מסלולים שונים בהשוואה בין סעיף זה לפלט מסעיף 19 נזהה ש הוא שונה לחלוטין,  
   בסעיף 19 קיבלנו ביחידות מרחק,  
   לאומת סעיף זה בו קיבלנו ביחידות כסף,  
   משום שנחברו מסלולים שונים ניתן גם לזהות בעמודת שהערכים מקיימים עבור סעיף 19 המרחק קטן יותר, ועבור סעיף זה המחיר נמוך יותר.*
3. היוריסטיקה MDATestsTravelTimeToNearestLabHeuristic עבור
4. *Sadf*
5. *Asd*
6. *Asd*
7. *Asd*
8. *Asd*
9. *Asd*
10. *Asd*