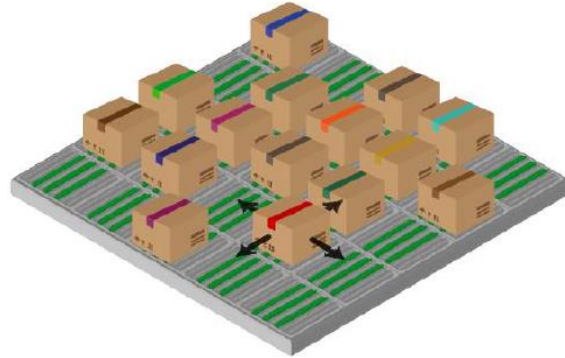


הפקולטה להנדסה ע"ש אייבי ואלדר פליישמן המחלקה להנדסת תעשייה



ליקוט פריטים במערכת אחסון המבוססת פאזל

תכן מפעלים

פרופ' יוסי בוקצ'ין

ד"ר אלון בלוד

תוכן עניינים:

א. הצגת המודל והנחותיו	2
ב. 1. תיאור מילולי של האלגוריתם	5
ב. 2. הדגמת יישום האלגוריתם על הוצאת פריט אחד	12
ג. טבלת סיכום של תוצאות האלגוריתם	16
ד. 1. סביבת עבודה	16
ד. 2. תוצרים	17

א. הצגת המודל והנחותיו

הצגת המודל:

מטרת המודל הינה להוציא את כלל הפריטים אשר נדרשים לצאת מן המחסן ביחידת זמן המועטה ביותר. את המודל שלנו ביססנו על המאמר של Gue & Kim 2007. חילקנו את המחסן לשלושה מחסנים קטנים יותר כאשר בכל מחסן ישנה נקודת יציאה אחת – תחת הנחות כלליות מספר 4 ניתן לראות את אופן החלוקה. כל פריט צריך לצאת מנקודת היציאה ששייכת לאזורו. לאורך המודל אפשרנו תנועה בו זמנית של הפריטים כאשר מספר הפריטים היכולים לבצע תנועה בו זמנית היא לכל היותר כמספר ה-escort הפנויים בתוספת למספר נקודות היציאה. כמו כן שמרנו על כך שבמידה ופריט זז בנקודת זמן מסוימת לא יוכל לנוע שנית בנקודת זמן זו.

המודל משייך לכל פריט, אשר מספרו מופיע ברשימת הפריטים להוצאה, escort אשר תפקידו ללוות את הפריט עד לנקודת היציאה. ההתאמה מתבצעת פעם אחת ע"י התייחסות למרחק הפריט מנקודת היציאה ומרחק הפריט ל escort הקרוב ביותר אליו. בחרנו לשייך באופן חד פעמי escort לפריט ולא לשנות את ה escort במהלך הדרך על מנת לשמור על העקרונות של Gue & Kim 2007 ולא לקטוע רצף של תנועות מסוג שלשות או חמישיות.

השלבים אשר escort עובר על מנת להוביל פריט אל נקודת היציאה הינם:

- תנועת escort אחת - ה- escort מתקדם בכל יחידת זמן לכיוון ה-target של הפריט בהתאם למיקום הרצוי, כפי שמתואר בהנחת אלגוריתם מספר 5. לאחר הגעתו, ה- escort והפריט מבצעים ביניהם חילוף מקומות ע"י צעד אחד נוסף.
- 3 תנועות escort - ה- escort מבצע צעדי שלשות עד אשר הפריט מגיע לשורה או לעמודה של נקודת יציאה. כל צעד מתוך השלוש מתבצע ביחידת זמן בודדת.
- 5 תנועות escort - ה- escort מבצע צעדי חמישיות עד אשר הפריט מגיע לנקודת היציאה. כל צעד מתוך החמש מתבצע ביחידת זמן בודדת.

כאשר הפריט בנקודת היציאה הוא מבצע חילוף עם נקודת היציאה – מה שמסמן את היציאה מהמחסן. החילוף עם נקודת היציאה מתואר בהנחת מודל מספר 7.

הנחות כלליות של המודל :

1. escort הינו אובייקט מסוג פריט שמספרו הוא 0. לכן כאשר נציין בדוח את המילה פריט זה יכול גם פריט מסוג escort.
2. על מנת להתאים את המודל לאופן שבו ה-python עובד הגדרנו כי העמודות במחסן הן בין 0-69 והשורות הן בין 0-29.
3. מיקום הפריט בנוי בצורה זו (j,i) כאשר j זה עמודה ו- i שורה.
4. המחסן מחולק לשלושה אזורים באופן הבא :
 - אזור a - בין 0-19 כאשר נקודת היציאה ממוקמת ב- $(4,0)$.
 - אזור b - בין 20-49 כאשר נקודת היציאה ממוקמת ב- $(34,0)$.
 - אזור c - בין 50-69 כאשר נקודת היציאה ממוקמת ב- $(64,0)$.
5. לכל אזור נספור את מספר ה-escort הפנויים כאשר האתחול הראשוני הינו 0.
6. נקודות היציאה מסומנות כפריט שמספרו 0. בנוסף, כאשר פריט יוצא הוא הופך ל-escort ולכן מספרו הופך ל-0.
7. כאשר מתבצעת יציאה של פריט מהמחסן, לדוגמא עבור פריט X שיצא בנקודת היציאה $(4,0)$ ביחידת זמן 3, פורמט הצעד יראה כך : $[3,[4,0,X],[4,0,0]]$. כלומר פריט שממוקם בנקודת היציאה יתחלף עם נקודת היציאה בעלת מספר 0.
8. יחידה הזמן הראשונה הינה 1.
9. לכל פריט שצריך לצאת מהמחסן משויך escort. לאורך המודל אנו מסתכלות על צעדים שה escort עושה במטרה לקדם את הפריט לנקודת היציאה.
10. כאשר escort משויך לפריט הוא מחובר אליו עד שהפריט יוצא מהמחסן או עד אשר פריט מסוג זה כבר לא צריך לצאת יותר. במקרה זה ה-escort אינו תפוס יותר ויכול להיות משויך לפריט אחר.
11. ייתכן מצב שבו פריט אשר משויך לו escort מסוים, יבצע תזוזה בעזרת escort אחר. ה-escort האחר יתחלף עם הפריט בדרך ליעד שלו. בנוסף ייתכן כי escort יוזז ע"י escort אחר.
12. תנועה של פריט יותר מפעם אחת ביחידת זמן אינה אפשרית. במידה ופריט הוזז ביחידת זמן לא יתכן כי פריט אחר יזוז שנית בנקודת זמן זו.

הנחות לאלגוריתם:

1. כלל המרחקים באלגוריתם חושבו לפי מרחקי מנהטן.
2. תכונת work לאובייקט מסוג פריט - תכונה שמציינת האם הפריט משויך ל- escort או לא. כאשר הערך free, הפריט פנוי כלומר ניתן לשייך לו פריט אחר מסוג escort. כאשר הערך busy, הפריט כבר שויך ל- escort אשר ילווה אותו עד נקודת היציאה. תכונה זו משמשת לצורך מעקב על מנת לראות כי לפריט יש escort שמלווה אותו עד ליציאה ול- escort יש פריט שמשויך לו ואינו ישויך לפריט אחר.
3. תכונת moving לאובייקט מסוג פריט - כאשר פריט כלשהו זז ביחידת זמן מסוימת בעקבות צעד שהוא מבצע או בעקבות צעד שפריט אחר ביצע איתו, נרשם לו הערך 1 אשר מעיד כי בוצעה תזוזה. במידה והוא זז הוא אינו יכול לבצע צעד נוסף באותה יחידת זמן. תכונה זו משמשת לצורך מעקב על מנת שפריט לא יזוז יותר מפעם אחת ביחידת זמן בודדת.
4. תכונת out לאובייקט מסוג פריט - הפריטים בעלי מספר אשר מופיע ברשימת הפריטים להוצאה, מקבלים את הערך True, כלומר צריך להוציא את הפריט. במידה ומספר פריט מופיע ברשימת הפריטים להוצאה כלל הפריטים במחסן בעלי מספר זה יסומנו ב True ללא תלות במספר הפעמים שהפריט צריך לצאת.
5. Target - המיקום שה- escort צריך להגיע ליד הפריט כדי להתחיל לבצע איתו צעדי שלישיות. ה- target נקבע בהתאם למאמר של Gue & Kim 2007 וביחס למיקום הפריט ולנקודת היציאה.
 - כאשר $i \geq j$ - target יהיה בשורה מתחת לפריט באותה עמודה.
 - כאשר $i < j$ - נבדוק את מיקום הפריט ביחס לנקודת היציאה.
 - במידה והפריט ממוקם מימין לנקודת היציאה - ה- target יהיה בעמודה משמאל לפריט באותה שורה.
 - במידה והפריט ממוקם משמאל לנקודת היציאה - ה- target יהיה בעמודה מימין לפריט באותה שורה.
 - במידה והפריט נמצא באותה עמודה של נקודת היציאה - ה- target יהיה שורה מתחת לפריט באותה עמודה.
 - מקרה ספציפי הוא כאשר הפריט נמצא בנקודה (0,0) אז ה- target יהיה בעמודה מימין לפריט באותה שורה.
6. תכונת stage לאובייקט מסוג פריט: ישנן 4 רמות אשר פריט יכול להיות בהן:
 - 0 - הפריט נמצא בנקודת היציאה
 - 1 - escort מתקדם בצעדים בודדים לעבר הפריט עד שיגיעה ל- target שלו ויבצע עם הפריט חילוף.
 - 3 - escort מבצע צעדי שלישיות על מנת לקדם את הפריט לנקודת היציאה.

- 5 - escort מבצע צעדי חמישיות על מנת לקדם את הפריט לנקודת היציאה.
- 7. לכל פריט שמגיע לפונקציה finding close escort ישויד escort.
- 8. האלגוריתם הינו חמדן והוא אינו מתייחס להשפעת הצעדים העתידיים שלו.
- 9. ישנן ארבע אופציות לכיוון הצעד הבא - למעלה, למטה, ימינה ושמאלה.

1.ב. תיאור מילולי של האלגוריתם

באלגוריתם ישנן שתי מחלקות :

1. מחלקה עבור כל פריט במחסן - לכל פריט ישנן התכונות הבאות :
 - מספר הפריט.
 - מיקום הפריט כאשר האתחול הראשוני הינו לפי מיקום הפריט אשר הופיע בקובץ הנתונים.
 - מרחק הפריט מנקודת היציאה כאשר האתחול הראשוני הוא מספר גדול מאוד.
 - מרחק הפריט מה - escort הקרוב ביותר כאשר האתחול הראשוני הוא מספר גדול מאוד.
 - תכונת out כמתוארת בהנחת אלגוריתם מספר 4.
 - האזור אליו הפריט שייך לפי המיקום שלו ונקודת היציאה הרלוונטית לפי הנחת מודל מספר 4.
 - תכונת work, כמתוארת בהנחת אלגוריתם מספר 2, מאותחלת להיות free.
 - תכונה ששומרת את האובייקט של escort המשויד לו, מאותחלת להיות None.
 - תכונה ששומרת את כיוון הצעד הבא של הפריט.
 - תכונת moving כמתוארת בהנחת אלגוריתם מספר 3, מאותחלת להיות 0.
 - תכונת stage כמתוארת בהנחת אלגוריתם מספר 6, מאותחלת להיות 1.
 - תכונה ששומרת את האובייקט פריט שממוקם במיקום שהינו שהצעד הבא של ה- escort.
2. מחלקה עבור המחסן - למחסן ישנן התכונות הבאות :
 - רשימה שמתארת את מיקום הפריטים ההתחלתיים במחסן ע"פ קובץ הנתונים.
 - רשימה המכילה את הפריטים שצריכים להוציא מן המחסן ע"פ קובץ הנתונים.
 - רשימה אשר מכילה את כלל האובייקטים מסוג פריט הקיימים במחסן - נבנתה ע"י קריאה למחלקת פריט.
 - הגדרת המיקומים של שלושת נקודות היציאה לפי הנחת מודל מספר 4.
 - step list - רשימה שמכילה את כלל הצעדים שבוצעו באותה יחידת זמן.

- תכונה ששומרת את יחידת הזמן הנוכחית.
- final list - רשימה מצטברת אשר בכל סוף יחידת זמן מוסיפה את step list יחד עם יחידת הזמן המתאימה. הרשימה הסופית אותה נציג בסוף ריצת הקוד.
- תכונה ששומרת כמה escort פנויים יש בכל אזור.
- רשימות אשר שומרות אילו פריטים צריך להוציא בכל אזור בהתאם לכמות ה- escort הפנויים באותו אזור.
- list need to go - רשימה אשר מכילה את כל הפריטים, שביחידת זמן מסוימת, שמבוצע עבורם צעד במטרה לקדם אותם לכיוון נקודת היציאה.
- final out list - רשימה שמכילה את כל סוגי פריטים שיצאו מן המחסן ואת יחידת הזמן בה התבצעה היציאה. הרשימה הסופית אותה נציג בסוף ריצת הקוד.

כלל הפונקציות הבאות נמצאות תחת מחלקת המחסן :

1. פונקציית function one : מטרת פונקציה זו היא להכין את הפריט שצריך לצאת. הפונקציה מוצאת לו את ה- escort הקרוב ביותר ומוצאת ל- escort את הצעד הבא אותו צריך לבצע. בנוסף פונקציה זו בונה את רשימת list need to go.

1.1. פונקציית which dist from io : פונקציה זו מוצאת, מכלל הפריטים, את אלו שתכונת out בעלת ערך True ושערך התכונה work הינה free. במידה והפריט עומד בתנאים אלו, הפונקציה שולחת אותו לפונקציה dist from IO אשר תחשב את המרחק בינו לבין נקודת היציאה השייכת לאזור אליו הוא מוגדר.

1.1.1. פונקציית dist from IO : פונקציה זו מחשבת את המרחק בין הפריט לבין נקודת היציאה השייכת לאזור אליו הוא מוגדר.

1.2. פונקציית exit item : פונקציה זו עוברת על כלל הפריטים ובודקת האם פריט בעל תכונת out שמכילה True וגם האם המיקום שלו שווה למיקום של נקודת היציאה. במידה וכן - תכונת stage של הפריט משתנה ל 0 והפריט נכנס לרשימת list need to go.

1.3. פונקציית count escort : פונקציה זו עוברת על כלל הפריטים ובודקת האם פריט הוא escort והאם הינו בעל תכונת work שמכילה free. במידה וכן - מוסיפה אותו לסכום ה- escort הפנויים שיש באזור בו הוא נמצא.

1.4. נבדוק האם יש לפחות escort פנוי אחד, באחד מן האזורים.

1.5. במידה ויש escort פנוי -

1.5.1. פונקציית finding min : פונקציה זו עוברת על כלל הפריטים ובודקת האם הפריט בעל תכונת out המכילה True, האם הינו בעל תכונת work שמכילה free והאם המיקום שלו אינו שווה למיקום נקודת היציאה. במידה וכן - פריט זה מתווסף לרשימה המתאימה

לאזור בו הוא נמצא. לאחר מכן, רשימות אלו ממוינות לפי המרחק מנקודת היציאה - מהמרחק הקטן ביותר עד הגדול. הפונקציה בונה רשימה לכל אזור שמכילה את הפריטים עם המרחק המינימלי. גודל הרשימה נקבע לפי מספר ה- escort הפנויים או מספר הפריטים שצריך להוציא - המינימלי מבניהם. שלושת רשימות אלו מאוחדות לרשימה אחת ונשלחות אל פונקציית finding close escort.

1.5.1.1. פונקציית finding close escort: פונקציה זו מוצאת לכל פריט אשר נשלח אליה את ה- escort הפנוי הקרוב ביותר ע"י שימוש בפונקציית dist from escort. ה- escort שנבחר נשמר כתכונה של הפריט - מה שיוצר את החיבור ביניהם. בנוסף, תכונת work שלהם משתנה ל- busy. במידה ואין escort פנוי באזור של הפריט, ישויד אליו escort מאזור אחר. לאחר מציאת escort לפריטים, נשלח אותם לפונקציית match.

1.5.1.1.1. פונקציית dist from escort: פונקציה זו מחשבת את המרחק בין הפריט לבין ה- escort.

1.5.1.1.2. פונקציית match: פונקציה זו מחשבת את הצעד הבא שעל ה- escort לבצע על מנת לקדם את הפריט לנקודת היציאה. תחילה נקבע ה- target של הפריט לפי הנחת אלגוריתם מספר 5.

- במידה וה- escort ממוקם כך שהוא והפריט כבר ביצעו חילוף (ה- escort נמצא מעל הפריט או מאחד מצדדיו) ולא נדרש לבצע החלפה נוספת, לא מוזן ל- escort צעד חדש.
- במידה וה- escort עדיין לא הגיע ל- target ירשם לו כי כיוון הצעד הבא שלו הוא בהתאם למה שיקרב אותו לכיוון ה- target :
 - במידה וה- escort - משמאל ל- target והצעד הבא שהוא צריך לעשות אינו הפריט שלו - ירשם לו כי כיוון הצעד הבא שלו הוא לנוע ימינה. במקרה והצעד הבא שהוא צריך לעשות הוא הפריט שלו, ימשיך לחפש צעד אחר.
 - במידה וה- escort - מימין ל- target והצעד הבא שהוא צריך לעשות אינו הפריט שלו - ירשם לו כי כיוון הצעד הבא שלו הוא לנוע שמאלה. במקרה והצעד הבא שהוא צריך לעשות הוא הפריט שלו, ימשיך לחפש צעד אחר.
 - במידה וה- escort בעמודה של ה- target וה- escort נמצא מתחתיו - ירשם לו כי כיוון הצעד הבא שלו הוא לנוע למעלה.
- במידה וה- escort נמצא מעליו - ירשם לו כי כיוון הצעד הבא שלו הוא לנוע למטה.

- במידה וה- escort הגיע ל- target ירשם לו כי כיוון הצעד הבא שלו הוא בהתאם למה שיגרום להחלפת מיקומים עם הפריט.
 - במידה וכיוון הצעד הבא של ה- escort ריק וכאשר גם הפריט וגם ה- escort נמצאים באותה השורה או העמודה של נקודת היציאה, תכונת stage תהיה 5. במידה והצעד הבא של ה- escort ריק, תכונת stage תהיה 3.
 - במידה והפריט ממוקם בנקודת היציאה, תכונת stage תהיה 0.
- 1.5.2. כל הפריטים בעלי תכונת work המכילה busy שהם אינם escort, נכנסים לרשימת list need to go.
- 1.6. אם אין escort פנוי - כל הפריטים בעלי תכונת work המכילה busy שהם אינם escort, נכנסים לרשימת list need to go.
- 1.7. עוברים על כל פריט ברשימת list need to go. לכל escort של פריט מהרשימה, בודקים מה מוזן לו ככיוון הצעד הבא ומשייכים לאובייקט escort תכונה המכילה אובייקט מסוג פריט שממוקם איפה שהצעד הבא שלו. פעולה זו מתרחשת רק בצעד הראשון של escort ששייך לפריט ברשימה.
- 1.8. עוברים לפונקציה function two.
2. פונקציית function two: מטרת פונקציה זו היא למצוא את הצעד הבא של כל escort המשוך לפריט ברשימת list need to go.
- במידה והפריט אינו ממוקם בנקודת יציאה הפריט נשלח לפונקציית ה- match הרלוונטית לפי הרמה שבה הוא נמצא:
- במידה ונמצא ברמה 1 - ישלח לפונקציה match כפי שתוארה בסעיף 1.5.1.1.2.
 - במידה ונמצא ברמה 3 - ישלח לפונקציה match for three.
 - במידה ונמצא ברמה 5 - ישלח לפונקציה match for five.
- * בצעד הראשון של ה- escort שמשויך לפריט הוא לא נשלח שנית לפונקציית match אלא נשלח ישר לסעיף 2.3.1.
- 2.1 פונקציית match for three: פונקציה זו בונה את הצעד הבא של ה- escort אשר מבצע שלישית צעדים עם הפריט.
- במידה והמרחק בין הפריט ל- escort גדול מ- 2 - ה- escort צריך להתמקם מחדש בעזרת פונקציית match.

בתרשימים הבאים נתאר כיצד ה- escort נע בהתאם למיקום הפריט ובהתאם למיקום הפריט ביחס לנקודת היציאה. בתרשים ניתן לראות את מיקום הפריט כאשר החצים מתארים כיוון תנועה של escort.

- כאשר הפריט ממוקם מימין לנקודת היציאה אך לא בשורה של נקודת היציאה:



* אם נמצאים בשורה מס' 0 כיוון הצעד הבא אינו נרשם כ - למטה אלא תכונת stage של הפריט הופכת ל- 5.

- כאשר הפריט ממוקם משמאל לנקודת היציאה אך לא בשורה של נקודת היציאה:



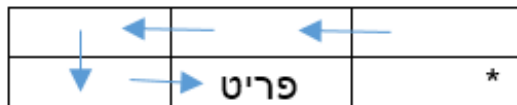
* אם נמצאים בשורה מס' 0 כיוון הצעד הבא אינו נרשם כ - למטה אלא תכונת stage של הפריט הופכת ל- 5.

- כאשר הפריט ממוקם בעמודה של נקודת היציאה:



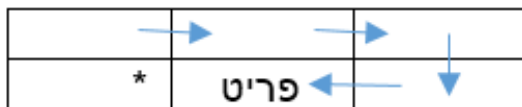
* כיוון הצעד הבא אינו נרשם אלא תכונת stage של הפריט הופכת ל- 5.

- כאשר הפריט ממוקם בשורה של נקודת היציאה ונמצא מימין לנקודת היציאה:



* כיוון הצעד הבא אינו נרשם אלא תכונת stage של הפריט הופכת ל- 5.

- כאשר הפריט ממוקם בשורה של נקודת היציאה ונמצא משמאל לנקודת היציאה:

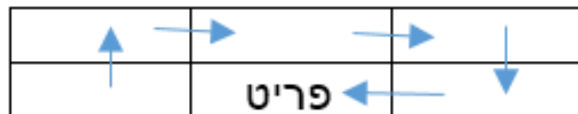


* כיוון הצעד הבא אינו נרשם אלא תכונת stage של הפריט הופכת ל- 5.

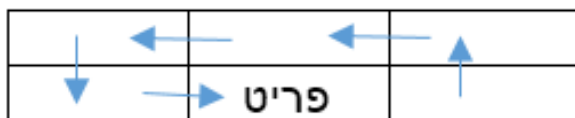
2.2. פונקציית match for five: פונקציה זו בונה את הצעד הבא של ה- escort אשר מבצע חמישיית צעדים עם הפריט.

בתרשימים הבאים נתאר כיצד ה- escort נע בהתאם למיקום הפריט ובהתאם למיקום הפריט ביחס לנקודת היציאה. בתרשים ניתן לראות את מיקום הפריט כאשר החצים מתארים כיוון תנועה של escort.

- כאשר הפריט ממוקם מצד שמאל של נקודת היציאה:



- כאשר הפריט ממוקם מצד ימין של נקודת היציאה:



- כאשר הפריט ממוקם באותה עמודה של נקודת היציאה:



2.3. לכל escort של פריט, בודקים מה מוזן לו ככיוון הצעד הבא ומשייכים לאובייקט escort תכונה המכילה אובייקט מסוג פריט שממוקם איהה שהצעד הבא שלו. לאחר שהוגדר ל- escort מה הצעד הבא שלו או כאשר הפריט נמצא בנקודת היציאה, הפונקציה שולחת את הפריט לפונקציית run direction.

2.3.1. פונקציית run direction: פונקציה זו מנווטת את ה-escort לפונקציית direction הרלוונטית לפי תכונת stage שלו.

- במידה וערך stage הינו 0 - ישלח לפונקציית exit.
- במידה וערך stage הינו 1,3,5 - ישלח לפונקציה direction.

2.3.1.1. פונקציית exit: מטרתה לבצע את הצעד אשר יוציא את הפריט מחוץ למחסן. הצעד שהתבצע בנקודת זמן זו, בין הפריט לבין נקודת היציאה, מתווסף ל- step list. תכונת out של הפריט הופכת ל- false. בנוסף מספר הפריט הופך ל- 0 כלומר הוא הופך ל-escort. תכונת work של הפריט ושל ה-escort המשוך אליו משתנה ל- free. כמו כן, תכונת הפריט ששומרת את אובייקט ה-escort מתאפסת. בנוסף, נבדק האם פריט נוסף בעל אותו מספר נמצא ברשימת הפריטים שצריכים לצאת מהמחסן. במידה ולא - לא נרצה שפריטים שכבר אינם צריכים לצאת, ימשיכו לנוע לכיוון היציאה. לכן, לפריטים הקיימים במחסן בעלי אותו מספר, מתעדכנת תכונת out ל- False. תכונת work של הפריט ושל ה-escort המשוך אליו משתנה ל- free. כמו כן, תכונת הפריט ששומרת את אובייקט ה-escort מתאפסת.

2.3.1.2. פונקציית direction: פונקציה זו בודקת האם צריך להתבצע צעד. תכונת moving של ה-escort ושל אובייקט הפריט, שמשויך לתכונה הצעד הבא של ה-escort, נבדקת. במידה והערך שווה 1 לאחד מהם - ה-escort לא יבצע צעד נוסף באותה יחידת זמן, כלומר לא יתקדם לפי הצעד שרשום לו. במידה והערך שווה 0 - אובייקט הפריט (שנמצא בצעד הבא) ואובייקט ה-escort נשלחים לפונקציית step. בנוסף, התכונה של ה-escort שמכילה את כיוון הצעד הבא מתאפסת.

2.3.1.2.1. פונקציית step: מטרת פונקציה זו היא לבצע צעד החלפה בין פריט ו-escort. הצעד שבוצע באותה יחידת זמן, מתווסף ל- step list. בנוסף, תכונת moving מתעדכנת לערך 1 אצל הפריט ואצל ה-escort. המיקום של הפריט והמיקום של ה-escort מתעדכנים בהתאם והפריט נשלח לפונקציית dist from IO המופיעה בסעיף 1.1.1 על מנת שיחושב לו המרחק החדש מנקודת היציאה.

2.4. הפונקציה שולחת לפונקציה final step.

2.4.1. פונקציית final step: פונקציה זו יוצרת את שתי הרשימות אותן עלינו להחזיר כפלט: final list ו- final out list. היא מוסיפה, לכל צעד שבוצע ביחידת זמן ושנמצא ב- step list, את יחידת הזמן הנוכחית, מאפסת את step list ומעדכנת את יחידת הזמן. בנוסף תכונת moving מתאפסת לפריט שהוזז ול-escort.

ב.2. הדגמת יישום האלגוריתם על הוצאת פריט אחד

8							
7		פריט 61					
6							
5							
4				escort1			
3							
2						escort2	
1							
0					I/O		
	0	1	2	3	4	5	6

ניקח לדוגמא את פריט 61 הצבוע בכחול באיור. הפריט ממוקם ב (1,7). פריט זה נמצא ברשימת הפריטים שצריך להוציא מהמחסן ולכן תכונת out מאותחלת להיות True. הפריט שייך לאזור a ונקודת היציאה הרלוונטית אליו הינה (4,0). שאר התכונות של הפריט מאותחלות להיות לפי האתחול הראשוני שמופיע בסעיף 1 בתיאור האלגוריתם.

function one מתחילה לרוץ. פונקציית which dist from io עומד מכיוון שפריט 61 עומד בתנאים המתוארים בסעיף 1.1 בתיאור האלגוריתם היא שולחת את פריט 61 יחד עם נקודת היציאה הרלוונטית לו לפונקציית dist from IO.

פונקציית dist from IO מחשבת את המרחק בין פריט 61 לנקודת היציאה. תכונת המרחק בין הפריט לנקודת היציאה מתעדכנת להיות 10.

לאחר מכן מופעלת פונקציית exit item. מכיוון שפריט 61 אינו עומד בתנאים המתוארים בסעיף 1.2 בתיאור האלגוריתם הפונקציה אינה משפיעה על הפריט.

פונקציית count escort מופעלת. באזור a נמצאים שני escort (צבועים בכתום באיור) שתכונת work שלהם מכילה את הערך free. מכיוון שקיימים escort פנויים, נעבור לפונקציית finding min.

בפונקציה זו הפריט יתווסף לרשימת פריטים שצריכים למצוא להם escort. הרשימה נשלחת לפונקציית finding close escort. פונקציה זו מחשבת בעזרת פונקציית dist from escort את המרחק בין הפריט לבין ה- escort הפנויים. escort1 הוא בעל המרחק המינימלי לפריט 61 ולכן ישוך אליו.

תכונת work של הפריט ושל escort1 משתנה ל- busy ואובייקט escort1 הנבחר משוּיך לפריט 61 ונרשם כתכונה שלו. לאחר מכן, פריט 61 מועבר לפונקציית match.

8							
7		פריט 61					
6		target					
5							
4				← escort1			
3							
2						escort2	
1							
0					I/O		
	0	1	2	3	4	5	6

ניתן לראות שפונקציית match מצאה את ה- target הצבוע באדום (ממוקם מתחת לפריט מכיוון ש $i = 7 > j = 1$) ואת כיוון הצעד הבא של escort1 - "שמאלה". פריט 61 עומד בתנאים המתוארים בסעיף 1.5.2 בתיאור האלגוריתם ולכן נכנס לרשימת list need to go.

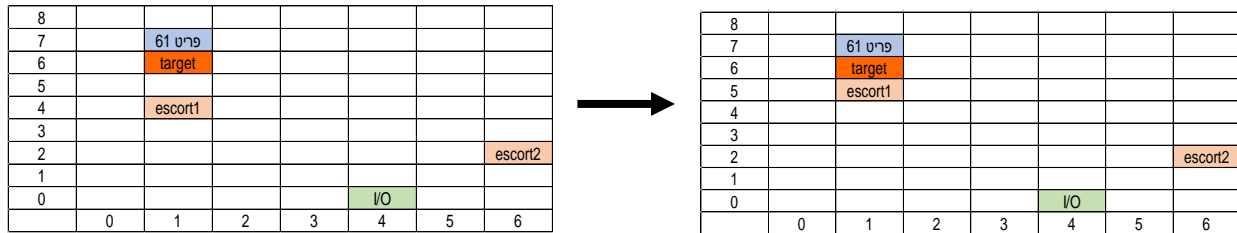
הצעד הבא של escort1 הינו (2,4) ולכן האובייקט שזהו המיקום שלו משויך ל- escort1 כתכונה של הצעד הבא.

function two מתחילה לרוץ. מכיוון שפריט 61 אינו עומד בתנאים המפורטים בסעיף 2 בתיאור האלגוריתם הוא נשלח לפונקציית run direction. מכיוון שתכונת stage של הפריט הינה 1, הוא נשלח יחד עם escort1 לפונקציית direction. מכיוון ש- escort1 עומד בתנאים המתוארים בסעיף 2.3.1.2 בתיאור האלגוריתם escort1 והפריט שמשויך לו כצעד הבא כלומר האובייקט במיקום (2,4), נשלחים לפונקציית step. כעת מתבצע הצעד הראשון - escort1 עובר להיות ממוקם ב (2,4). צעד זה מצטרף לרשימת step list. תכונת moving מתעדכנת לערך 1.

8							
7		פריט 61					
6		target					
5							
4			escort1				
3							
2							escort2
1							
0					I/O		
	0	1	2	3	4	5	6

באזור ניתן לראות את המיקום החדש של escort1. לאחר מכן מופעלת פונקציית final step אשר מאפסת את תכונת moving ל- escort. כעת חוזרים חזרה לפונקציית function one. הפריט נכנס לרשימת list need to go. עוברים לפונקציית function two, מכיוון שפריט 61 עומד בתנאים המפורטים בסעיף בתיאור האלגוריתם 2, ובגלל שה- stage שלו הינו 1 הוא נשלח לפונקציית match. לאחר שנקבע לו כיוון לצעד - "שמאלה", התכונה של הצעד הבא של escort1 מתעדכנת להיות אובייקט מסוג פריט שמיקומו (1,4). כעת, פריט 61 נשלח לפונקציית run direction. מכיוון שתכונת stage שלו הינה 1, הוא נשלח יחד עם escort1 לפונקציית direction. מכיוון ש- escort1 עומד בתנאים המתוארים בסעיף 2.3.1.2 בתיאור האלגוריתם escort1 והפריט שמשויך לו כצעד הבא כלומר האובייקט במיקום (1,4), נשלחים לפונקציית step. כעת מתבצע הצעד - escort1 עובר להיות ממוקם ב (1,4). צעד זה מצטרף לרשימת step list. תכונת moving מתעדכנת לערך 1. התהליך חוזר על עצמו, לפי המסלול המתואר באזור, עד אשר escort1 מגיע ל target ומבצע חילוף עם פריט 61:

8							
7		פריט 61					
6		target					
5							
4			← escort1				
3							
2							escort2
1							
0					I/O		
	0	1	2	3	4	5	6



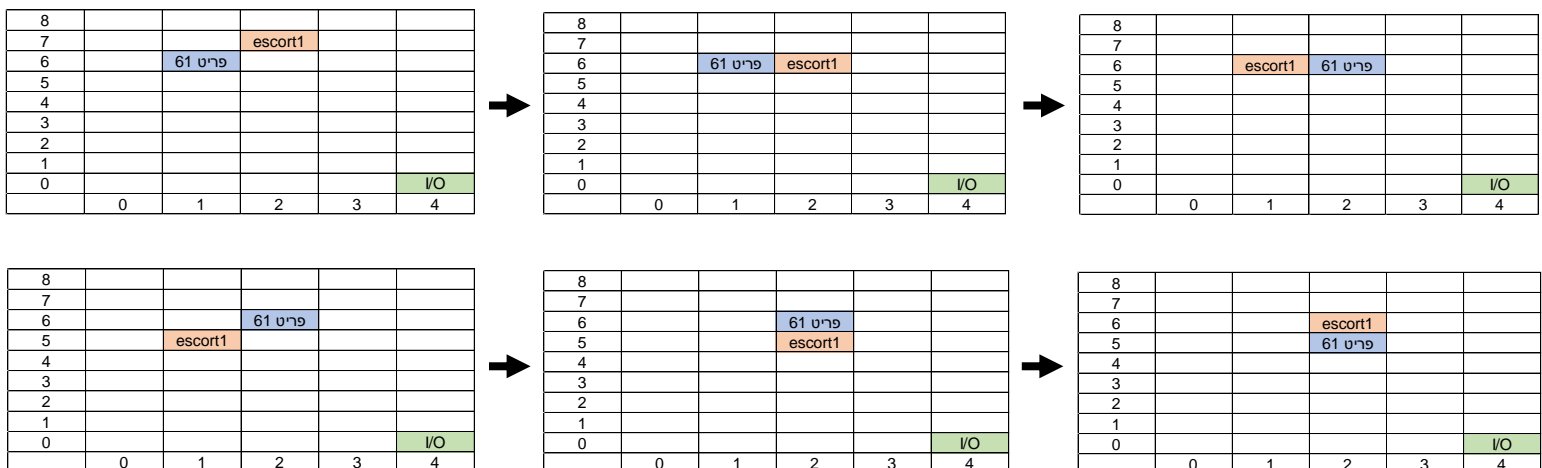
המצב העדכני הינו :

8							
7							
6							
5							
4							
3							
2							
1							
0							
	0	1	2	3	4	5	6

כעת משתנה תכונת stage של הפריט והערך הינו 3, לפיכך ניתן להתחיל בצעדי שלשות. כתוצאה מהשינוי ערך ה- stage ל- 3, ישלח הפריט לפונקציית match for three. כעת יתבצעו צעדי שלשות לפי אותו התהליך שתואר עד כה.

8							
7							
6							
5							
4							
3							
2							
1							
0							
	0	1	2	3	4	5	6

באיור ניתן לראות את צעדי השלשות שבוצעו : שחור ← תכלת ← כתום ← צהוב ← ירוק ← סגול. מצורפים איורים המתארים את זוג צעדי השלשות הראשון שבוצע.



המצב שמתקבל לאחר כל צעדי השלשות הינו :

8							
7							
6							
5							
4					escort1		
3					פריט 61		
2							escort2
1							
0					I/O		
	0	1	2	3	4	5	6

במצב זה כאשר הפריט ו- escort1 ממוקמים בעמודה של נקודת היציאה ו- escort1 ממוקם מעל הפריט, תכונת stage של הפריט משתנה לערך 5 וניתן להתחיל בצעדי חמישיות. כתוצאה משינוי הערך של תכונת stage ל- 5, ישלח הפריט לפונקציית match for five. כעת יתבצעו צעדי חמישיות לפי אותו התהליך שתואר עד כה.

8							
7							
6							
5							
4					escort1		
3					פריט 61		
2							escort2
1							
0					I/O		
	0	1	2	3	4	5	6

באזור ניתן לראות את צעדי החמישיות שבוצעו : שחור ← כתום ← סגול.

מצורפים איורים המתארים את צעד החמישיות הראשון שבוצע.

8						
7						
6						
5						
4						escort1
3					פריט 61	
2						
1						
0					I/O	
	0	1	2	3	4	5



8						
7						
6						
5						
4						
3					פריט 61	escort1
2						
1						
0					I/O	
	0	1	2	3	4	5

8						
7						
6						
5						
4						
3					פריט 61	
2						escort1
1						
0					I/O	
	0	1	2	3	4	5



8						
7						
6						
5						
4						
3					פריט 61	
2						escort1
1						
0					I/O	
	0	1	2	3	4	5

8						
7						
6						
5						
4						
3					escort1	
2					פריט 61	
1						
0					I/O	
	0	1	2	3	4	5

בסיום צעדי החמישיות התקבל המצב הבא :

8							
7							
6							
5							
4							
3							
2						escort2	
1					escort1		
0					פריט 61		
	0	1	2	3	4	5	6

ניתן לראות כי הפריט ממוקם בנקודת היציאה. כעת, תכונת stage של הפריט משתנה לערך 0 והוא נשלח לפונקציית exit. פונקציית exit מבצעת את היציאה. צעד היציאה יראה כך : $[t, [4, 0, 61], [4, 0, 0]]$ כאשר t מציג את יחידת הזמן שבה בוצעה הוצאת הפריט 61 מהמחסן. פונקציה זו משנה את מספר פריט 61 להיות 0 כלומר escort נוסף מצטרף. הפונקציה מאתחלת את תכונות work, out והתכונה שמחזיקה את אובייקט ה-escort. במידה ו-61 אינו מופיע יותר כפריט שצריך להוציא אותו, לכלל פריטי 61 המופיעים במחסן, התכונות שמפורטות לעיל יתעדכנו בהתאם.

ג. טבלת סיכום של תוצאות האלגוריתם עבור הקלטים השונים המצורפים באתר

מספר escort	100	50	10
מספר יחידות הזמן	471	440	379

ד.1. סביבת עבודה

בוצע שימוש בסביבת עבודה - python 3.7.

לא בוצע שימוש בחבילות חיצוניות.

לא נדרשות פעולות מיוחדות על מנת להריץ את האלגוריתם (יש לוודא שתחת Project Interpreter מופיעים pip ו-setuptools).

האלגוריתם פועל עם קבצי pickle שמות זהים לשמות הקבצים אשר ניתנו בפרויקט.

בתחילת האלגוריתם ניתן לראות כי מבוצע ייבוא של כלל קבצי pickle אל האלגוריתם. כמו כן, בסיומו ניתן לראות שמבוצעות שלוש הרצות של האלגוריתם - הרצה לכל מספר escort (הרצה עבור escort 10, הרצה עבור 50 והרצה עבור 100). לבסוף, מתקבלים שישה קבצי pickle כאשר לכל מספר escort משויכים שני קבצים, כמפורט בהמשך.

ד.2. תוצרים

בתיקייה מצורפים מספר קבצים :

- שישה קבצי pickle נתונים המכילים את הפריטים במחסן ואת הפריטים הנדרשים להוצאה.
- קובץ python המכיל את האלגוריתם ומוכן להרצה.
- קובץ pdf המכיל את הדוח.
- תיקיית output המכילה שישה קבצי pickle. לכל מספר escort התחלתי מופיעים שני הקבצים הבאים :
- X_escorts_steps.p - כאשר X הינו מספר ה-escort ההתחלתיים. קובץ זה מכיל רשימה של כלל הצעדים שבוצעו בתוספת יחידת הזמן.
- X_escorts_items_out.p - כאשר X הינו מספר ה-escort ההתחלתיים. קובץ זה מכיל רשימה של סוגי הפרטים שהוצאו מהמחסן בתוספת יחידת הזמן שההוצאה בוצעה.