

טופס הצעת פרויקט – י"ג הנדסת תוכנה

MiniMovesCube MMC



חתימת מנחה

חתימת משרד החינוך

חתימת הסטודנט





Table of Contents

1Type chapter title (level 1)

2Type chapter title (level 2)

3Type chapter title (level 3)

4Type chapter title (level 1)

5Type chapter title (level 2)

6Type chapter title (level 3)



רקע תיאורטי בתחום הפרוייקט

הקובייה ההונגרית (3X3) הומצאה לראשונה בשנת 1974 על ידי הפסל והפרופסור ההונגרי "ארנה רוביק" ("Erno Rubik") ונקראה על ידו "קוביית הקסם". הקובייה ההונגרית שווקה לראשונה בשנת 1980 כצעצוע משחק ונקראה מחדש על ידי שם הממציא – קוביית רוביק. אם כי ברוב שפות העולם היא גם ידועה בשם זה, בשפה העברית היא קרויה על שם מדינת המוצא של הממציא. הקובייה נחשבת לצעצוע הנמכר ביותר בהיסטוריה עם מספר מכירות של כ-300 מיליון קוביות. מאז ועד היום נוצרו גרסאות רבות לקובייה הפשוטה כך שכל פאה מחולקת למספר שונה של ריבועים ואף נוצרו דגמים בצורות גיאומטריות שונות.

קובייה הונגרית היא למעשה פאזל מכני העשוי פלסטיק שכל אחת משש פאותיה בצבע שונה ומחולקת למספר ריבועים שווים, היכולים לנוע ולהחליף מקומות אלו עם אלו. הקובייה ההונגרית המוכרת והפופולרית ביותר היא קובייה בגודל של 3X3, כלומר כל פאה בה מחולקת לתשעה ריבועים שווים.

במהלך פרוייקט זה נעסוק בקובייה ההונגרית שתוארה לעיל (3X3) ללא צורך בציון מחדש.

במהלך השנים פתרונות כללים רבים התגלו באופן עצמאי כאשר הפתרון הפופולרי ביותר פותח על ידי דייוויד סינגמסטר. פופולריותו נובעת בשל קלותו להבנה אנושית ולכן נחשב לפתרון נאיבי שאין באמת בכוונתו ליעל את הפתרון בכל צורה שהיא, הן מבחינת זמן והן מבחינת מספר הצעדים הנדרשים בפתרון. הפתרון הכללי עובד כך:

פתירת הפאה העליונה ← פתירת השכבה האמצעית ← פתירת הפאה התחתונה

בשלב מאוחר יותר פותחו אלגוריתמים שמטרתם היא לפתור את הקובייה מהר ככל האפשר – פתרונות אלה מכונים פתרונות בזק. הנפוץ ביותר מביניהם הוא פתרון שהומצא על ידי ג'סיקה פרדריך. שיטה זו יעילה מאוד לפתירת הקובייה בצורה של שכבה אחר שכבה ודורשת אלגוריתמים רבים. כמו כן עוד פתרון בזק פופולרי הינו פתרון שנוצר על ידי לארס פטרוס, המובסס על בניית פינה אחת ופיתוחה ללא השלמת אף פאה (בניגוד לפתרונות נאיביים הקלים לעין האנושית) עד מספר סיבובים מועט מאוד לקראת הפתרון הסופי.





בשנת 2007 דניאל קונקל וג'ין קופרמן ניסו להשתמש במחשב על מנת להראות שניתן לפתור קובייה מכל מצב אפשרי במקסימום של 26 מהלכים. בשל כך שלקובייה יש מיליארדי ווריאציות אפשריות, המחשב תוכנת להביא לאחד מ-15,752 מצבים כאשר כל אחד מהם יכול להיפתר תוך כמה מהלכים בודדים נוספים. כל הפתרונות הוכחו כפתירים ב-29 מהלכים, כאשר רובם ניתנים לפתרון ב-26 מהלכים. הפתרונות שהובאו על ידי המחשב ביותר מ-26 מהלכים נפתרו אחר כך במפורש על מנת להוכיח את טענתם כי ניתן לפתור קובייה מכל מצב אפשרי במקסימום של 26 מהלכים.

בשנת 2008 תומאס רוקיקי הוכיח בצורה חישובית שניתן לפתור כל מצב אפשרי של הקובייה במקסימום צעדים של 25. מאוחר יותר באותה השנה הוא הכריז על הוכחה ל-22 מהלכים.

לבסוף, בשנת 2010, הוכיחו תומאס רוקיקי, הרברט קוצ'ימבה, מורלי דוידסון וג'ון דטרידג' את ההוכחה הסופית, בעזרת מחשב, כי ניתן לפתור את כל עמדות הקוביות עם מקסימום של 20 מהלכים. מספר זה מכונה המספר של אלוהים.

אלגוריתם זה מכונה "אלגוריתם המספר של אלוהים" ובאנגלית "God's Number Algorithm". הידוע גם כ "God's Algorithm" והינו בר מימוש רק בעזרת מחשב. דרך הפעולה של האלגוריתם היא גישה של מיקום כל ריבוע בהתאמה תוך התחשבות בשאר הקוביות – כך נוצר שבעבור כל ריבוע בודד נצטרך לעבור על כל שאר הריבועים ולכן למרות שהפתרון הינו הכי יעיל מבחינת כמות צעדיו הוא דווקא לא הכי יעיל מבחינת זמן הריצה שלו במחשב.

בחרתי לעשות את הפרוייקט על נושא הקובייה ההונגרית במטרה לפתור כל קובייה שתתקבל במינימום מהלכים. בחרתי בנושא זה משום שניסיתי לאתגר את עצמי מבחינה אלגוריתמית ככל הניתן וכל נושא שעלה במוחי היה מייד לי מידי מבחינת הפתרון האלגוריתמי שלו.

ואז ראיתי בחדרי את הקובייה ההונגרית ומייד הרעיון בער בי והתחיל להתפתח. כשקראתי מאמרים בנושא על מנת להבין למה אני נכנס הבנתי שהפרוייקט הזה מציע בדיוק את האתגר האלגוריתמי ששאר הפרוייקטים שחשבתי עליהם לא ידעו להציע. כמו כן, אני אוהב מאוד קוביות הונגריות על כל סוגיהן ויש לי אוסף מכובד של קוביות ואף צורות גיאומטריות שונות וכמובן שאני יודע לפתור אותן, כך שיש לי חובה אישית לנושא וביערה פנימית שלא נראה לי שהייתי מוצא בשום פרוייקט אחר.

בנוסף לכך על מנת להפוך את הפרוייקט ליותר ייחודי, אנסה להשתמש בתורה הגדולה שנאספה עד היום בנושא על מנת לייצר אלגוריתם משלי, כך שאנסה ליעל את זמן הריצה של התוכנית ככל האפשר בעזרת שימוש במבני נתונים שונים בשאיפה לשמור על מינימום של מהלכים עד לפתירת הקובייה.



תיאור הפרוייקט

המשתמש יקבל פריסה של קובייה הונגרית ויציב בה את הצבעים בהתאם לקובייה שברצונו לפתור.

בהמשך, אנסה לשנות את תצורת בניית הקובייה על ידי המשתמש כך שהקובייה תהיה תלת מימדית ועליה יבחר את הצבעים.

מטרת הפרוייקט היא לקבל קובייה הונגרית, לוודא את תקינותה ורק לאחר מכן, תציג את הצעדים לפתרונה כך שנשאף למספר מהלכים מינמלי, כלומר 20 מהלכים, כפי שאלגוריתם המספר של אלוהים שתואר בחלק התיאורטי יודע לעשות.

כדי לעשות זאת נצטרך לממש את אלגוריתם המספר של אלוהים. האלגוריתם קשה מאוד למימוש וסדר גודל זמן הריצה שלו גדול מאוד, כפי שהוסבר בחלק התיאורטי - מאחר והוא דורש מעבר על כל הריבועים בעבור כל ריבוע יחיד.

בעקבות זאת, אנסה ליעל את זמן הריצה ותוך כדי זה לא תהיה ברירה אלא להגדיל את מספר הצעדים המינמלי של פתרון הקובייה. ועל כן, נגדיר מחדש את מטרת הפרוייקט - כך:

לקבל קובייה הונגרית, לוודא את תקינותה ורק לאחר מכן, תציג את הצעדים לפתרונה כך שנשאף למספר מהלכים מינמלי תוך התחשבות בזמן הריצה.

מימוש פרוייקט זה דווקא דורש אלגוריתם חדש שדרך פעולתו שונה מהותית מדרך הפתרון של האלגוריתם של אלוהים. ניישם גישה הוליסטית יותר הנקראת ירידה דרך תת קבוצות מקוננות.

על מנת ליעל את זמן הריצה ולממש את הגישה של ירידה דרך תת קבוצות מקוננות אשתמש במבני הנתונים הבאים: גרפים ועצים.



הגדרת הבעיה האלגוריתמית

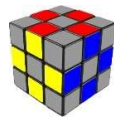
קודם כל נדרש אלגוריתם לבדיקת תקינות הקובייה, אותו נכנה אלגוריתם הבדיקה.

לצורך הבנת האלגוריתם נצטרך להגדיר מספר מושגים:



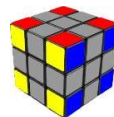
ריבוע אמצעי:

ממוקם באמצע אחת מהפאות בהתאמה



ריבוע פנימי:

ממוקם באמצע שורה או טור בפאה בהתאמה



ריבועים פינתיים:

ממוקם בצד שורה או הכי

למעלה/למטה בטור בפאה בהתאמה

אלגוריתם הבדיקה:

1. אם אין 6 ריבועים אמצעיים כך שכל אחד מהם שונה בצבעו מהאחר, החזר שקר. אחרת, המשך אל צעד 2.
2. אם אין 4 ריבועים פנימיים מאותו הצבע, החזר שקר. אחרת, המשך אל צעד 3.
3. אם בעבור כל אחד מהריבועים הפנימיים, מי שילדו מהפאה הסמוכה הינו בצבע המתאים, המשך אל צעד 4. אחרת, החזר שקר.
4. אם אין 4 ריבועים פינתיים מאותו הצבע, החזר שקר. אחרת, המשך אל צעד 5.
5. אם בעבור כל אחד מהריבועים הפנימיים, מי שילידו משתי הפאות הסמוכות הינו בצבע המתאים, החזר אמת, אחרת החזר שקר.

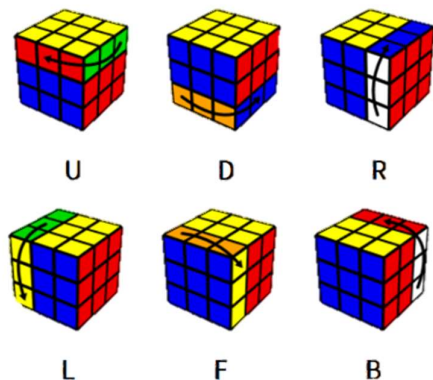
כדי לפתור את הבעיה האלגוריתמית שנותרה, פתירת קובייה הונגרית תקינה במינימום צעדים, בגישה ההוליסטית הנקראת ירידה דרך תת קבוצות מקוננת יש צורך בלמצוא תת קבוצה מן הקבוצה הנוכחית (שבהתחלה היא הקובייה המתקבלת, לאחר בדיקת תקינותה) ובעבורה למצוא את קבוצת המשנה שרצף ההזות בה, שיהפוך כל איבר בקבוצה המשנה לחלק מתת הקבוצה, הוא הקטן ביותר. נכנה את אלגוריתם זה כאלגוריתם הפתירה.





כדי להבין את הפתירה נצטרך קודם להגדיר את הסימונים הבסיסיים בקובייה ההונגרית.

סימונים אלה מוגדרים בצורה הבאה:



- (Front) F - מייצג את סיבוב פני הקובייה עם כיוון השעון
- (Right) R - מייצג את סיבוב צידה הימני של הקובייה עם כיוון השעון
- (Left) L - מייצג את סיבוב צידה השמאלי של הקובייה עם כיוון השעון
- (Up) U - מייצג את סיבוב הצד העליון של הקובייה עם כיוון השעון
- (Down) D - מייצג את סיבוב הצד התחתון של הקובייה עם כיוון השעון
- (Back) B - מייצג את סיבוב הצד האחורי של הקובייה עם כיוון השעון

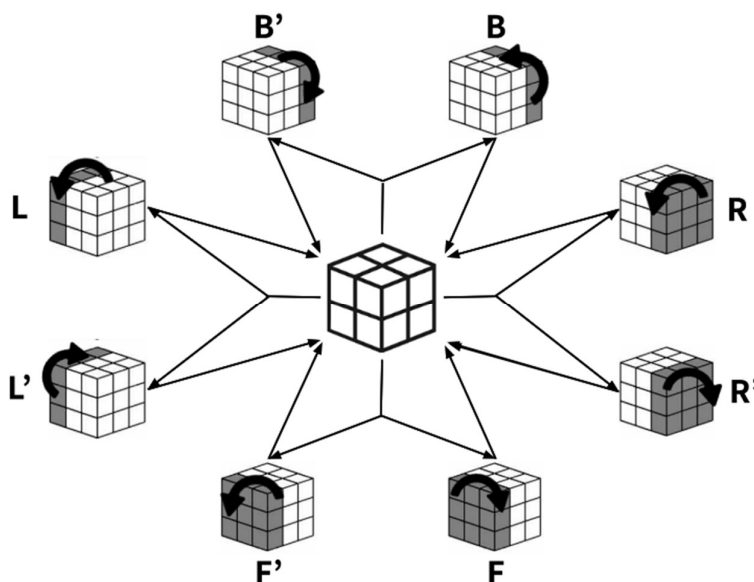
הסימן ' (לדוגמא F') מייצג נגד כיוון השעון (לדוגמא F' מייצג את סיבוב פני הקובייה נגד כיוון השעון).

הסימן 2 (לדוגמא F2) מייצג 2 סיבובים (ולכן זה לא משנה גם לאיזה כיוון נסובב משום שהקובייה היא סימטרית, 3X3).

לאחר שהגדרנו את הסימונים נוכל להגדיר את המושג הזזות פנים כך:

F, R, L, B אך לא D ו-U

איור לתיאור הליך הזזות הפנים בצורת גרף:





אלגוריתם הפתירה:

1. נסמן את הקובייה כקבוצה G .
2. בחר קבוצה של הזזות של פנים הקובייה כך שתיווצר תת קבוצה מתאימה: H (כך שתת הקבוצה יכולה להיות בהזזות פנים בלבד) שבעבורה $|G| / |H|$ הוא הקטן ביותר.
3. בעבור כל קבוצת משנה של H , מצא את רצף ההזזות הקטן ביותר שיהפוך כל איבר בקבוצת המשנה לחלק מתת הקבוצה H ונסמן אותו ב E .
4. אם $H = E$ אז עצור, אחרת בצע:
1.4. $G \leftarrow H$.

למעשה, עתה סיימנו וכעת נדרש רק להציג את הפיתרון שמצא אלגוריתם הפתירה.

יש לשים לב שלא נגיע לחלק של אלגוריתם הפתירה כל עוד אלגוריתם הבדיקה מחזיר שקר, כלומר כאשר הקובייה אינה פתירה.

ולכן זימון האלגוריתמים יראה כך:

אם אלגוריתם הבדיקה מחזיר אמת, המשך אל אלגוריתם הפתירה. אחרת, החזר הודעת שגיאה, שהקובייה אינה תקינה.



תהליכים עיקריים

בניית הפריסה של הקובייה ואת ממשק המשתמש, מימוש האלגוריתמים: הבדיקה והפתרון והצגת הפתרון למשתמש.

שפת תכנות

C#

סביבת עבודה

Visual Studio 2022

לוח זמנים

מטרות	תאריך יעד סיום
בניית פריסת הקובייה	01.02.2023
מימוש אלגוריתם הבדיקה	01.03.2023
מימוש אלגוריתם הפתרון	01.04.2023
בניית ממשק המשתמש	01.05.2023