

הפקולטה למדעי המחשב הטכניון, מכון טכנולוגי לישראל

ארגון ותכנות המחשב

234118

: תרגיל מספר

|  |  |  |
| --- | --- | --- |
| תא להחזרה: | 12/6 | תאריך הגשה: |

מגישים

Shiransaada@campus

.technion.ac.il 3 0 1 7 3 1 9 9 8 שירן סעדה

דואר אלקטרוני מספר ת .ז. שם מלא

[raudnitz@campus.technion](mailto:raudnitz@campus.technion) 3 0 1 7 2 9 5 4 7 אסף ראודניץ

.ac.il

דואר אלקטרוני מספר ת .ז. שם מלא

ציון

תיעוד חיצוני תרגיל 3:

פונקציות שנדרשו:

**calcWP**

רגיסטרים:

R0- אינדקס לתנועה על הלוח

R1- מכיל את החיבור של האינדקס R0 והכתובת של Board בעזרתו רואים את תוכן המשבצת הנוכחית האם מכילה 0,1,2.

R4- מכיל ערך את הערך של WP שאותו מחשבים בפונקציה

Sp- מצביע למחסנית, משתמשים בה לקרוא לפונקציות נוספות ולשמור ולהעביר מידע להן.

Pc- רגיסטר קישור לפונקציות שנקראות מתוך calcWP

אופן הפעולה:

הפונקציה שומרת את ערכי r0,r1 ומכניסה למחסנית שני מקומות לחישוב, לספירת כמות החיילים הלבנים והשחורים בהתאמה.

לאחר מכן הפונקציה קוראת לפונקציה שעוברת על הלוח ומגדילה באחד את המקום המתאים במחסנית לפי הערך של המשבצת הנוכחית בלוח.

לאחר חישוב מספר החיילים מכל סוג, אם אין חיילים ליריב מחזירים ניצחון, אחרת אם אין חיילים לשחקן הנוכחי מחזירים הפסד, אחרת בודקים את מספר הצעדים החוקיים בלוח, אם יש צעדים חוקיים מחזירים את ערך החיסור המתאים למספר החיילים של השחקן הנוכחי פחות מספר החיילים של היריב. אם אין יותר צעדים במגרש, מוחזר ניצחון עם ההפרש בין חיילי הנוכחי גדול מהפרש היריב והפסד אם ההפך.

**checkWin**

רגיסטרים:

R4- לשמירת ערך calcWP

Sp- מצביע למחסנית

Pc- רגיסטר קישור לפונקציות פנימיות

אופן הפעולה:

הפונקציה מזינה במחסנית את ההכנות הדרושות לקריאה לcalcWP ומקבלת ממנו חזרה ערך בR4, אם הערך הוא 20 הפונקציה תחזיר בתא ההחזרה שהוקצה לה במחסנית ניצחון, אחרת תחזיר שאין ניצחון.

**minOrMax**

רגיסטרים:

R4- להחזרת התוצאה

אופן הפעולה:

הפונקציה משווה בין השחקן הנוכחי והשחקן המקורי שהועברו במחסנית, במידה והם שווים תחזיר 1, אחרת תחזיר 1-.

**movPiece**

רגיסטרים:

R5- רגיסטר הקישור, בעזרתו מועבר תוכן inline

Sp- מצביע למחסנית

אופן הפעולה:

הפונקציה מקבלת את ערך המשבצת הנוכחית וערך משבצת היעד, עבור על אחד היא מכפילה את הערך ב2 (ע"מ שיתאים למימדי הלוח שבנוי ממילה כלומר 2 בתים לכל "תא") ומחברת אליו את כתובת הלוח.לאחר מכן מוסיפה מקום במחסנית לחישוב בו מיוצר ההפרש בין היעד למקור, לפיו ניתן לדעת עם החייל מנסה לזוז למעלה או למטה ומכך להסיק אם הוא לבן או שחור את הערך מכניסים לתווית של השחקן הנוכחי ומעוברים לצעד התאים לפי ההפרש שקיבלנו בין היעד למקור, לכל צעד של קפיצה יש סט פעולות משלו בו מוזן ביעד ערך השחקן המתאים ומנוקים המשבצות של המקור ושל הדילוג. עבור כל הצעדים שמזיזים שחקן בלי אכילה ישנו קטע קוד אחד שמכניס ביעד את השחקן הנוכחי ובמקור מוחק את הערך שהיה.

**getMaxWp**

מזינה את השחקן הנבחר בשחקן הנוכחי ואת הצעדים בצעדים הנוכחי וקוראת לתחילת הריצה.

לפני היציאה מכניסה את הערכים של wp,dst,src למחסנית

פונקציות נוספות:

**MovRec**

רגיסטרים:

R0- אינדקס לתנועה על הלוח

R1- מכיל את החיבור של האינדקס R0 והכתובת של Board בעזרתו רואים את תוכן המשבצת הנוכחית האם מכילה 0,1,2.

R2- רגיסטר קישור לפונקציות פנימיות

R3- מחזיק את ערך minOrMax לאורך כל הפונקציה

R4,R5- לחישובים (div,mul)

Sp- מצביע למחסנית

Pc- רגיסטר קישור לפונקציות פנימיות

אופן הפעולה:

הפונקציה מתחילה מלחשב ערך minOrMax ולפיו מזינה -21/21 לתוך WinParam, אם מחפשים מקסימום נזין ערך יותר קטן מכל תוצאה אפשרי, אם מחפשים מינימום נזין ערך יותר גדול מכל תוצאה אפשרית.

לאחר מכן הפונקציה רצה על כל הלוח, כאשר נמצאה משבצת המכילה את השחקן הנוכחי הפונקציה בודקה כל מהלך חוקי שהשחקן הנוכחי יכול לבצע, במידה ונמצא צעד חוקי של השחקן, הפונקציה תשמור את הצעד הנ"ל ואת הWP הנוכחי ותקרא לmovPiece ע"מ להזיז את החייל לפי הצעד החוקי שמצאה, לאחר שהחייל זז הפונקציה תקרא לרקורסיה עבור הלוח במצב החדש.לאחר החזרה מהרקורסיה הפונקציה תקרא לUndoMov ע"מ להחזיר את הצעד שבוצע חזרה, ולאחר מכן תשווה בין הWP שחזר מהרקורסיה לבין הWP ששמרה במחסנית לפני הקריאה לרקורסיה.

אם הWP השמור מהעבר הוא יותר טוב, אז הפונקציה תעדכן את הWP חזרה למה שהיה בעבר ותשחרר את השמירה של הצעד. לעומת זאת אם הפונקציה מוצאת WP יותר אז הWP מעודכן לערך הגבוה ביותר והפונקציה תשמור את הצעד הנוכחי שהוביל למצב האופטימלי הנ"ל (השמירה תתבצע רק אם מספר הצעדים הנוכחי שווה למקורי בכדי למנוע סיבוכים).לאחר השלמת על הצעד הנ"ל הפונקציה תמשיך לשאר הצעדים החוקיים עבור השחקן הנוכחי ולאחר מכן תמשיך לעבור על כל הלוח לחפש עוד שחקנים שיכולים לבצע מהלך חוקי ותקרא לרקורסיה בהתאם.

אם אין מהלכים חוקיים כלל עבור שחקן מסויים, הפונקציה תבדוק את מספר הצעדים הנוכחי, אם יש עוד צעדים אז הפונקציה תקרא לרקורסיה להמשך המשחק, אם אין עוד צעדים הפונקציה תחשב calcWP ותחזור חזרה עם הערך הנ"ל ללא צעדים.

**getAux**

רגיסטרים:

R2- שמירה של תוצאות מפונקציות אחרות

R4,R5- לחישובים (mul)

Sp- מצביע למחסנית

Pc- רגיסטר קישור לפונקציות פנימיות

אופן פעולה:

תחילה הפונקציהבודקת האם יש עוד מהלכים, אם אין עוד מהלכים נתקפל, אם יש עוד מהלכים בספירה הפונקציה בודקת אם אנחנו במצב של ניצחון עבור השחקן הנוכחי, אם כן נתקפל, אחרת נקרא לmovRec להמשך הרקורסיה. במצב של התקפלות כלומר הגענו לעלה בעץ, נבדוק את הWP ע"י הכפלת הערך מcalcWP בערך מminOrMax כך נקבל את הגודל של המהלכים בסימן הנכון מבוצע ע"י calcMOM. לאחר מכן נחזור חזרה.

**getMax**

פונקציית מעטפת לgetAux ששומרת רגיסטרים ומשנה שחקן נוכחי וכמות צעדים.

**undoMov**

רגיסטרים:

R5- רגיסטר הקישור, בעזרתו מועבר תוכן inline

Sp- מצביע למחסנית

אופן הפעולה:

הפונקציה מקבלת את אותם ערכים כמו movPiece ומבצעת בדיוק את הפעולה ההפוכה לו עבור אותו מקור ויעד.

**NumOfMov**

רגיסטרים:

R0- אינדקס לתנועה על הלוח

R1- מכיל את החיבור של האינדקס R0 והכתובת של Board בעזרתו רואים את תוכן המשבצת הנוכחית האם מכילה 0,1,2.

R2- רגיסטר קישור לפונקציות פנימיות

R4,R5- לחישובים (div)

Sp- מצביע למחסנית

אופן הפעולה:

זהו הקוד מתרגיל 2, מבצע סריקה של כל הלוח ובכל נקודה בה נמצא חייל זהה לחייל הנוכחי הפונקציה בודקת את כל המהלכים האפשריים בשבילו.

מחזירה ערך לתוך תווית NumMoves של כמות המהלכים החוקיים עבור צבע מסויים של שחקן בכל המגרש.

**chkMv**

קוראת ל NumOfMov עבור שני השחקנים ע"מ לחשב את כל המהלכים החוקיים בלוח

**SwapPly**

מחליפה את השחקן הנוכחי מ2 ל1 ולהפך

**calcMOM**

מעדכנת את הWP, הפונקציה בודקת את זוגיות הצעדים במגרש ומחליפה את הWP אם עבור צעד זוגי אם התוכן שהוזן לה גדול מWP אז הפונקציה מעדכנת את WP. ואם הצעד אי זוגי הפונקציה מעדכן את WP אם הוא קטן מהערך שהפונקציה קיבלה בr5.

**EvenOdd**

מעבירה את כמות הצעדים שנותרה לr5 ומחשבת האם המספר זוגי או אי זוגי ומדליקה את הדגל אם ההכמות זוגית