

הנושא: **אלגוריתמיקה - מה זה בכלל?**

הוכן ע"י: דוד הראל, מכון ויצמן.

תקציר: המאמר הוא פרק מהספר: "אלגוריתמיקה, יסודות מדעי המחשב", מאת דוד הראל. בפרק זה מוסבר מהו אלגוריתם, מה כולל אלגוריתם ומהו דורשים בפתרון בעיה אלגוריתמית - כל זאת תוך מתן דוגמאות לאלגוריתמים. בפרק זה מוצגת האלגוריתמיקה כליבו של מדע המחשב ומודגשת חשיבותה. כן מובאת סקירה היסטורית של התפתחות מדעי המחשב.

מילות מפתח: מדעי המחשב, אלגוריתם, היסטוריה של מדעי המחשב, היסטוריה של המתמטיקה, אוקלידס, אל-חואריזמי, ז'אקאר, באבג', הולרית'.

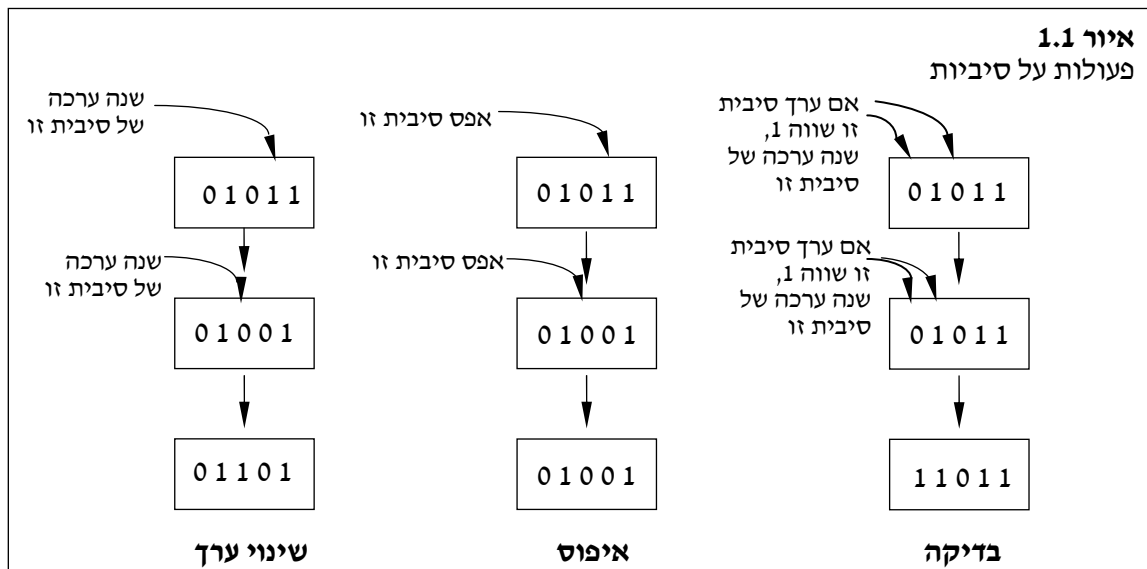
החומר פורסם במסגרת: על"ה 13, אלול תשנ"ג, ספטמבר 1993, עמודים 7-14.

החומר מכיל בנוסף לעמוד הפתיחה: 11 עמודים.

אלגוריתמיקה - מה זה בכלל?*

מחשבים הינם מכונות מופלאות. נדמה לנו שהם מסוגלים לעשות הכל. הם מטיסים מטוסים וספינות חלל, הם מפקחים על תחנות כוח ועל מפעלים כימיים מסוכנים. אי-אפשר עוד לנהל חברות בלעדיהם. מספר הולך וגדל של הליכים רפואיים מורכבים לא ניתנים כלל לביצוע בהעדרם. מחשבים משרתים עורכי דין ושופטים המחפשים תקדימים משפטיים באינספור משפטים מתועדים, והם מסייעים למדענים בביצוע חישובים מתמטיים מסובכים ומורכבים ביותר. מחשבים מנתבים ומפקחים על מיליוני שיחות טלפון המועברות דרך רשתות ביניבשתיות, ואת כל המטלות הם מבצעים בדיוק עצום - החל מקריאת מפות וסידור אותיות לדפוס וכלה בעיבוד תמונות גרפי ותכנון מעגלים משולבים. מחשבים פוטרם אותנו ממטלות משעממות, כגון מעקב קפדני אחר הוצאות הבית, ואף מספקים לנו בידור, למשל בדמות משחקי מחשב ומוסיקה ממוחשבת. יתר על כן, המחשבים של היום מנוצלים גם בתכנון המחשבים החזקים יותר של מחר. לאור דברים אלה, הרי זה מפתיע ואף מרשים ביותר שניתן לראות את המחשב הדיגיטלי - אפילו את המודרני והמורכב ביותר - כאוסף גדול של מתגים ותו לא. מתגים אלה, הנקראים **סיביות** (bits), יכולים להימצא באחד משני מצבים. במילים אחרות, כל סיבית יכולה להיות בעלת אחד משני **ערכים**, 0 או 1. בדרך כלל נקבע ערכה של סיבית על ידי מאפיין אלקטרוני כלשהו. למשל, ערך הסיבית מציין אם נקודה מסוימת טעונה מטען חשמלי חיובי או שלילי. מחשב יכול לבצע ישירות רק מספר קטן של פעולות פשוטות ביותר: שינוי ערכה של סיבית, איפוסה או בדיקת ערכה. שינוי ערך הסיבית משנה את ערכה מ-0 ל-1 ולהפך, איפוס נותן לסיבית את הערך 0, בדיקה מבצעת פעולה אחת אם יש לסיבית ערך 0 ופעולה אחרת אם יש לסיבית ערך 1 (ראה איור 1.1).

מחשבים יכולים להיות שונים זה מזה בגודלם (כלומר, במספר הסיביות), בסוגי הפעולות היסודיות שהם יכולים לבצע, במהירות שבה מבוצעות פעולות אלה, בתווך הפיסי המכיל את הסיביות והקובע את ארגון הפנימי, וגם בסביבתם החיצונית. פריט אחרון זה פירושו ששני מחשבים הזהים לגמרי בתפקודיהם, יכולים להיראות שונים מאוד זה מזה: אחד עשוי לדמות למכשיר טלוויזיה בעל לוח



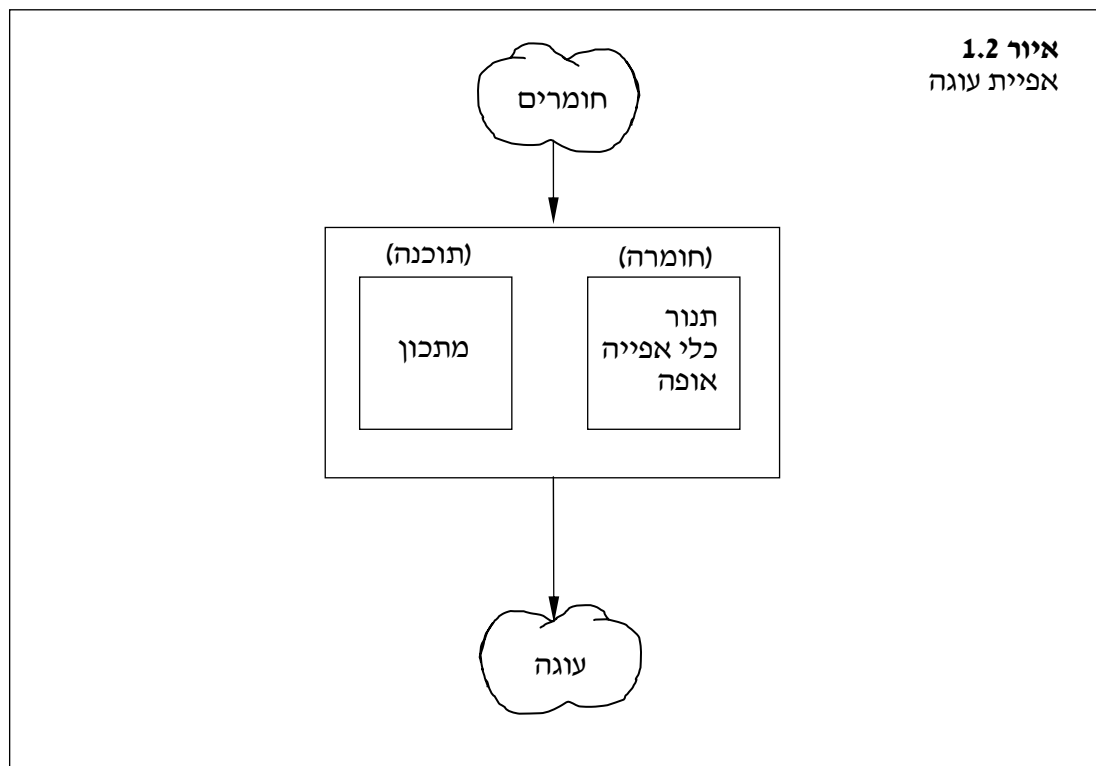
* מאמר זה הוא פרק מהספר "אלגוריתמיקה, יסודות מדעי המחשב" מאת דוד הראל. הספר הוא הגרסה העברית של הספר Algorithmics, A Spirit of Computing מאת דוד הראל. התירגום נעשה ע"י תמר אלמוג, בהוצאת האוניברסיטה הפתוחה, 1991. הפרק מתפרסם באדיבות המחבר וההוצאה.

מקשים, בעוד שהאחר ספון מתחת לכפתורים ולחוגות של מכונת סריגה אוטומטית. אולם, חשיבותה של התצורה החיצונית שולית בהשוואה לחשיבותם של הסיביות ושל ארגון הפנימי. הסיביות הן אשר חשות את הגירויים המגיעים מן העולם החיצון באמצעות כפתורים, ידידות, מקשי מקלדת, קווי תקשורת אלקטרוניים ואפילו מיקרופונים ומצלמות. הסיביות הן המחליטות כיצד להגיב לגירויים אלה, והן מגיבות בהתאם באמצעות הפניית גירויים אחרים מצידן אל צגים, מסכים, מדפסות, רמקולים וכדומה.

איך הם עושים את כל זה, המחשבים? מהו הדבר ההופך פעולות כל כך פשוטות על סיביות לאותם מעללים רבי רושם שמבצעים המחשבים לנגד עינינו? התשובה טמונה בשני המושגים המרכזיים בספר זה: **תהליך** (process), ו**אלגוריתם** (algorithm) הגורם לתהליך להתרחש והמכוון את מהלכו.

גסטרונומיה על קצה המזלג

תארו נא לעצמכם מטבח המכיל מאגר של חומרים, ערכת כלי אפייה, תנור אפייה ואופה (אנושי). אפייתה של עוגת צימוקים ערבה לחד הינה תהליך שמבצע האופה מן החומרים ובעזרתו של התנור, על פי מרכיב חשוב ביותר - המתכון. החומרים הם **הקלטות** (inputs) של התהליך, העוגה היא **הפלט** (output) והמתכון הוא **האלגוריתם**. האלגוריתם קובע אפוא אילו פעולות מהוות את התהליך. באופן כללי נקראים המתכונים, או האלגוריתמים, בשם **תוכנה** (software), בעוד שכלי האפייה והתנור מייצגים את **החומרה** (hardware). את האופה ניתן לראות במקרה זה כחלק מן החומרה (ראה ציור 1.2).



כמו במקרה של הסיביות והפעולות עליהן, גם להרכב אופה/תנור/כלי-אפייה יש יכולת ישירה מוגבלת מאוד. חומרת אפיית עוגה זו מסוגלת למזוג, לערבב, לשטוח, לטפטף, להדליק את התנור האפייה, לפתוח את דלת התנור ולמדוד זמן או כמויות, אולם אין היא יכולה לאפות באופן ישיר. המתכונים הם אלה שבכוחם להפוך כבמטה קסם את היכולות המוגבלות של חומרת המטבח לעוגות טעימות, והם אפוא הנושא המרכזי של ספר זה.

ובכן, מתכונים נקראים אלגוריתמים, ואילו את תחום המחקר, הידע והמומחיות האנושית המתמקדים באלגוריתמים נכנה בשם **אלגוריתמיקה** (algorithmics). ההקבלה ניתנה בצורה מדויקת ככל האפשר: המתכון, שהוא במובן מסוים ישות מופשטת, הוא האלגוריתם; הגרסה הכתובה פורמלית של המתכון, כזאת שאפשר למצאה בספרי בישול, מקבילה ל**תכנית מחשב** (computer program). במושג תוכנה כוונתנו למעשה לא לאלגוריתמים עצמם אלא לתכניות - ייצוגים מדויקים של אלגוריתמים, אשר כתובים בשפות מיוחדות שמחשב יכול לקרוא אותן. אף על פי כן, עד שנדון בתכניות מחשב בפרק 3, אין בהבחנה זו חשיבות רבה.

בכל אשר נפנה פוגשים אנו אלגוריתמים. תהליכים יומיומיים רבים מתבצעים על פי אלגוריתמים: החלפת גלגל נקור, בניית ארון, סריגת סוודר, חילוק מספרים, חיפוש מספר בספר הטלפונים, עדכון רשימת הוצאות או מילוי טופס מס-הכנסה. חלק מתהליכים אלה (חילוק למשל) מתקשרים בתודעתנו למחשבים יותר מאשר תהליכים אחרים (כגון בניית ארון), אולם הבחנה זו איננה חשובה כאן. כמו כן, למרות שמחשבים הם מעמודי התווך שעליהם נשען נושא דיוננו בספר זה, לא נתרכז כלל ועיקר בהיבטים הפיסיים שלהם, למעט התייחסות עקיפה בפרקים 3 ו-9. אנו מתעניינים כאן ברוחם של המחשבים; במתכונים המפעילים אותם, באלגוריתמים שלהם.

אלגוריתמיקה לעומת מדעי המחשב

אלגוריתמיקה היא יותר מאשר סתם ענף של מדעי המחשב. היא לבו של מדע המחשב, והיא רלוונטית למעשה גם לרוב ההיבטים של מדע, כלכלה וטכנולוגיה. עצם טבעה של האלגוריתמיקה הופך אותה ישימה במיוחד באותם תחומים הנתמכים על-ידי מחשבים, ואלה הופכים במהירות לרוב מוחץ.

אנשים שואלים לפעמים: מה הם למעשה מדעי המחשב? מדוע לא קיימים מדעי הצוללת, מדעי המגהץ או מדעי הטלפון? שהרי טלפונים ומגהצים חיוניים לחיינו המודרניים לא פחות מאשר מחשבים; אולי אפילו יותר. שאלה מנקודת ראות שונה היא זו: האם אין הנושאים הנדונים במדעי המחשב כלולים כבר בתחומים מדעיים קלאסיים כגון מתמטיקה, פיסיקה, הנדסת חשמל, בלשנות, לוגיקה ופילוסופיה?

ובכן, ספר זה אינו מנסה לענות על שאלות אלה. תקוותנו היא שהספר יבטא משהו מן הייחודיות ומן האוניברסליות של האלגוריתמיקה, ובכך ישפוך אור גם על חשיבותם של מדעי המחשב כתחום מחקר אוטונומי, אם כי צעיר. הואיל ומחשבים עלולים להגביל במידה ניכרת את כלליות האלגוריתמיקה, רואים אנשים מסוימים את הקשר הבלתי נמנע בין השניים כעניין מצער. כבר נאמר שכינוי התחום בשם "מדעי המחשב" דומה להתייחסות לכירורגיה כאל "מדעי הסכין". בכל מקרה, ברור שאלגוריתמיקה לא הייתה נראית כפי שהיא נראית כיום ללא הקשר הזה. אף על פי כן קיימת הסכמה כללית שהמונח "מדעי המחשב" הוא שם מטעה. שמות מעין "מדעי המידע" או "מדעי התהליך" עשויים להיות מוצלחים יותר. שוב, כל שאנו טוענים הוא שהנושא שלנו, אלגוריתמיקה, מהווה את היסוד שעליו נשענים מדעי המחשב, ואין הוא בא להחליף אותם.

לכמה מן הנושאים שבהם נדון בהמשך, כגון קיומן של בעיות שאינן ניתנות לפתרון באמצעות מחשב, יש השלכות פילוסופיות הנוגעות לא רק למגבלות של אותן מכוונות נפלאות שאנחנו מסוגלים לבנות, אלא גם למגבלות שלנו כיצורים בני תמותה בעלי מסה ואורך חיים סופיים. מבלי להתעלם מאופיים העמוק של השלכות אלה, הדגש בספר זה הוא על יעד מעשי יותר: רכישת הבנה עמוקה לגבי מהותם של התהליכים הניתנים לביצוע על-ידי מחשב ושל המתכונים, האלגוריתמים, המפקחים עליהם.

קצת היסטוריה

הבה נסקור עתה כמה אבני דרך חשובות בהתפתחות מחשבים ומדעי המחשב, בעיקר כדי להראות שכתחום מדעי מסודר, הרי זה תחום צעיר ביותר.

בזמן מן הזמנים בין השנים 400 ל-300 לפני הספירה, המציא המתמטיקאי היווני הדגול אויקלידס (Euclid) אלגוריתם לחישוב המחלק המשותף המקסימלי (ממ"מ) של שני מספרים שלמים חיוביים. הממ"מ של X ושל Y (שני מספרים שלמים חיוביים) הוא המספר השלם החיובי הגדול ביותר שמחלק, ללא שארית, הן את X והן את Y . למשל, הממ"מ של 80 ושל 32 הוא 16. פרטיו של אלגוריתם זה אינם מעניינים כאן, אולם **האלגוריתם האויקלידי**, כפי שהוא נקרא, נחשב לאלגוריתם הלא-טריוויאלי הראשון שפותח אי פעם.

המילה אלגוריתם גזרה משמו של המתמטיקאי הפרסי מוחמד אל-חואריזמי (al-Khwarizmi), שחי במאה התשיעית לספירה ונחשב כמי שניסח לראשונה את כללי החיבור, החיסור, הכפל והחילוק השיטתיים של מספרים עשרוניים. אויקלידס ואל-חואריזמי היו, לא ספק, אלגוריתמיקנים לעילא ולעילא.

נפנה עתה מן התוכנה אל החומרה. אחת המכונות הראשונות אשר ביצעה תהליך שניתן לומר עליו שפוקח על-ידי אלגוריתם, היה נול אריגה שהומצא ב-1801 על-ידי צרפתי בשם ז'וזף ז'אקאר (Jacquard). דוגמת האריגה נקבעה על-ידי כרטיסים שנוקבו במקומות שונים. מנגנון מיוחד חש את הנקבים, פיקח על בחירת החוטים ועל פעולות אחרות של המכונה. מעניין לציין שלנול של ז'אקאר לא היה דבר וחצי דבר עם המשמעות המספרית הצרה של המונח "חישוב".

אחת הדמויות החשובות והססגוניות ביותר בהסטוריה של מדעי המחשב היה צ'רלס באבג' (Babbage). מתמטיקאי אנגלי זה בנה באופן חלקי, בשנת 1833, מכונה שקרא לה "מכונת הפרשים", וזו חישה נוסחאות מתמטיות מסוימות. לאחר מכן הגה ותכנן באבג' מכונה מרשימה שקרא לה "המכונה האנליטית". בניגוד למכונת הפרשים שתוכננה לבצע מטלה מסוימת, היתה המכונה האנליטית אמורה לבצע אלגוריתמים, או תכניות, מקודדות על-ידי המשתמש כנקבים בכרטיסים. לו הייתה המכונה האנליטית נבנית, הייתה המקביל המתמטי לנול האריגה של ז'אקאר, שהיה למעשה מקור ההשראה שלה. מובן מאליו שהמכונה של באבג' היתה מכנית באופיה, עשויה מנופים, צירים, ידיעות וגלגלי שיניים; לא אלקטרוניקה וסיליקון. אף על פי כן, הרעיונות שבאו לידי ביטוי באופן תכנונה של המכונה האנליטית מהווים את הבסיס למבנה הפנימי ולפעולה של המחשבים דהיום. מקובל לחשוב שבאבג' הקדים בהרבה את זמנו, ורעיונותיו זכו להערכה אמיתית רק מאוחר הרבה יותר.

מהנדס אמריקני בשם הרמן הולרית' (Hollerith) המציא מכונה המבוססת גם היא על כרטיסים מנוקבים, ששימשה את המשרד האמריקני לרישום אוכלוסין כדי לאגור את תוצאות המפקד הלאומי שנערך בארה"ב ב-1890. אולם המחשב הראשון לשימוש כללי נבנה רק בשנות ה-40 של המאה ה-20, הן בתגובה לצרכים החישוביים של פיסיקאים ושל אסטרונומים והן כתוצאה טבעית מזמינותם של המכשירים האלקטרוניים והאלקטרומכניים הדרושים. באופן אירוני, מלחמת העולם השנייה - על פיתוח הפצצות ופיתוח הצפנים שהיו כרוכים בה - תרמה אף היא. כמה מדמויות המפתח בתקופה מכרעת ומלהיבה זו היו האנגלי אלן טיורינג (Turing), האמריקנים הווארד אייקן (Aiken), ג'ון מוצ'לי (Mauchly), ג'י. פרספר אקרט (Eckert) והרמן גולדשטיין (Goldstine), והגרמני-אמריקני המפורסם ג'ון פון נוימן (von Neumann).

הבה נחזור לתוכנה ולאגוריתמיקה. באמצע שנות ה-30, של המאה ה-20 נעשתה עבודה תיאורטית חשובה ביותר שהניחה את הבסיס לתיאוריית האלגוריתמים, ושתוצאותיה נוגעות ליכולות ולמגבלות של אלגוריתמים שניתן לבצעם על-ידי מחשב. ראוי לציין שעבודה זו, שחלקים ממנה נתאר בפרקים הבאים של ספר זה קדמה למימושו למעשה של המחשב. אף על פי כן, חשיבותה אוניברסלית ומתמשכת. כמה מדמויות המפתח בנושא זה, מתמטיקאים כולם, היו אלן טיורינג הנזכר לעיל, הגרמני קורט גדל (Gödel), הרוסי אנדריי א. מרקוב (Markov) והאמריקנים אלונזו צ'רץ' (Church), אמיל פוסט (Post) וסטפן קליני (Kleene).

שנות ה-50, וה-60, היו עדות להתפתחויות טכנולוגיות מהירות ומרחיקות לכת בתכנון מחשבים ובבנייתם. ניתן לייחס אותן לתחילתו של עידן המחקר הגרעיני וחקר החלל מצד אחד, ולפריחתם של עסקים גדולים ושל בנקים וכן לפעילותן המגוונת של ממשלות מצד שני. ניבוי מדויק של תופעות שונות ברמת הגרעין דרש כוח חישובי גדול מאוד, וכזה נדרש גם בתכנון ובהדמיה של משימות חלל.

חקר החלל דרש גם שיפורים בתקשורת הנתמכת על-ידי מחשבים, כדי שיקלו על ניתוח וסינון אמינים של נתונים, ואף כדי לשפר את איכות הנתונים המשודרים אל ומאת לוויינים וספינות חלל. עסקים, בנקים ופעילויות ממשלתיות דרשו מחשבים לצורך אחסון, אחזור וטיפול במידע הנוגע למספר גדול מאוד של אנשים, של פריטי מלאי, של נתונים כספיים וכיוצא באלה.

עדות מעניינת לחשיבותן של ההתפתחויות הטכנולוגיות הנוגעות למכונות עצמן בתקופה ההיא, ניתן למצוא בשמות של חברת המחשבים בעולם - IBM - International Business Machines, ושל אחד מן הארגונים המקצועיים הגדולים בעולם הקשורים למחשבים, ה- The Association for Computing Machinery - ACM. השם הראשון נטבע ב-1920, ואילו השני בשנות ה-40, המאוחרות. בשני השמות נגזרת M מן האות הראשונה במילה Machine - מכונה. (IBM התפתחה מחברה שנוסדה ב-1896 על-ידי הרמן הולרית' שהוזכר לעיל, לצורך ייצור מכונות החישוב שהמציא לרגל המפקד).

מדעי המחשב הוכרו כתחום אקדמי עצמאי באמצע שנות ה-60, בערך, כאשר נפתחו מחלקות למדעי המחשב בכמה אוניברסיטאות. ב-1968 פרסם ה-ACM המלצה לגבי תכנית לימודים במדעי המחשב שנתקבלה על רבים. תכנית זו מהווה את הבסיס לרוב תכניות הלימודים הנהוגות לתואר ראשון, והיא מעודכנת אחת לכמה שנים. כיום קיימת מחלקה למדעי המחשב, או מגמה למדעי המחשב בתוך המחלקה למתמטיקה או להנדסת חשמל, כמעט בכל מוסד אקדמי. בשנות ה-60 התעורר מחדש העניין בעבודות שנעשו בשנות ה-30 בתחום האלגוריתמיקה, ומאז ועד היום נעשים בו מחקרים חשובים רבים.

לא נתעכב עוד על המצב הטכנולוגי העכשווי: מחשבים נמצאים כיום בכל מקום. ילדים מבקשים ומקבלים מחשבים אישיים כמתנת יום הולדת; בכמה אוניברסיטאות נדרש כל סטודנט למדעי המחשב לרכוש לעצמו מחשב לצורך הכנת שיעורי בית; אין כמעט פעילות תעשייתית, מדעית או מסחרית המתנהלת ללא עזרתם של מחשבים.

דיכוטומיה מוזרה

למרות כל שנאמר לעיל (או אולי כתוצאה מכך), נחלק הציבור בכל הנוגע להכרת המחשב באופן מוזר. יש אנשים שאינם יודעים ולא כלום על מחשבים, אך ישנם גם כאלה השייכים לקהילה הולכת וגדלה של המבינים בהם - קהילה הכוללת ילדים בני שנים עשרה שנה שיש להם מחשב אישי בבית, וכן מספר הולך וגדל של אנשים העושים שימוש יומיומי במחשבים, כגון מנהלים, מהנדסים, בנקאים, טכנאים, וכמובן מתכנתים מקצועיים, מנתחי מערכות ועובדים בתעשיית המחשבים עצמה.

מדוע זה מוזר? ובכן, הנה לפנינו מדע שלגביו יש בינינו כאלה שאינם יודעים דבר, בעוד שמספר הולך וגדל של אנשים יודעים כביכול הכל! למעשה, מה שבאמת יוצא דופן וראוי לתת עליו את הדעת, הוא העובדה שחלקים גדולים וחשובים של מדעי המחשב אינם ידועים דיים, לא רק לבני הקבוצה הראשונה אלא גם לאלה מן הקבוצה השנייה.

אחת ממטרותיו של ספר זה היא לנסות ולהאיר פן אחד חשוב של מהפכת המחשבים על-ידי הצגתם של כמה מושגי יסוד, תוצאות ומגמות שהם בבסיסם של מדעי המחשב. הספר מכוון הן למי שהתחום זר לו והן למומחה. אם אין לך עדיין ידע בנושא זה, תלמד כאן על "רוחם" של מחשבים ועל אופיה של המחשבה המושקעת במציאת הדרך להפיק מהם את המרב, אך יהיה עליך לחפש במקום אחר חומר בכל הנוגע ל"גשמיותם". ברם, אם הנך בעל ידע בתחום, לא תמצא אולי חידוש רב בפרקים הראשונים של הספר, אך תוכל להפיק תועלת (כך אנו מקווים) משאר הפרקים.

כמה מגבלות של מחשבים

לפני שנצא למסענו, הבה נעמיד כנגד הפסקה שבה פתחנו את הפרק, כמה משימות שאותן אין המחשבים של היום מסוגלים עדיין לבצע. נדון בהן ביתר הרחבה בפרק המסיים את הספר, שבו נעסוק בקשר שבין מחשבים לבין בינה אנושית.

ובכן, המחשבים של היום מסוגלים לנתח בן רגע כמות עצומה של נתונים מתוך צילומי רנטגן רבים של מוח אדם. ניתוח זה משמש את המחשב ליצירת תמונה המשקפת חתך רוחב של המוח, המספר מידע על מבנה הרקמות שם, ובכך מאפשר להצביע על המיקום המדויק של תופעות חריגות כגון גידולים או עודף נוזלים. בניגוד מדהים לכך, לא קיים כיום מחשב שיכול לנתח תמונה יחידה, רגילה, של פניו של אותו אדם על מנת לקבוע את גילו בדיוק, נאמר, של עשר שנים. אך הרי כמעט כל ילד בן שנים-עשר שנה יכול לעשות זאת! מדהימה אף יותר היא יכולתו של תינוק בן שנה לזהות את פני אמו בתמונה שלא ראה מעולם, יכולת שהיא מן המחשבים והלאה (ואין זה רק מפני שלמחשבים אין אמהות...).

מחשבים מסוגלים לפקח, בדרך המדויקת והיעילה ביותר שאפשר להעלות על הדעת, על רובוטים תעשייתיים מתוחכמים ביותר המשמשים לבניית חלקי מכוניות מורכבים מאוד המכילים מאות רכיבים. לעומת זאת, המחשב המתקדם ביותר של היום אינו מסוגל להנחות רובוט בבניית קן של ציפור מערמת זרדים וזלזלים, משימה שכל ציפור בת שנים-עשר חודשים תבצע בהצלחה וללא קושי!

מחשבים של היום מסוגלים לשחק שחמט ברמה של אמן בינלאומי, כלומר לנצח למעשה את הרוב המכריע של השחקנים האנושיים. אולם, אם נשנה את כללי המשחק שינוי קטן ביותר (למשל, על-ידי שנרשה לפרש שני מהלכים בכל פעם, או על-ידי שנגביל את מהלכי המלכה לחמש משבצות), לא יוכל המחשב הטוב ביותר מבין אלה להסתגל לשינוי מבלי שבני אנוש יתכנתו או יבנו אותו מחדש. בניגוד לכך, שחקן שחמט חובב בן שנים-עשר שנה יוכל לשחק משחק טוב למדי בכללים החדשים, לאחר תרגול מועט מאוד, ואף ילך וישתפר עם הזמן.

הבדלים אלה מקורם בשוני שבין בינה אנושית לבינה ממוחשבת. נוכל לדון בעניינים אלה מעמדה טובה יותר בפרק 12, לאחר שנלמד עוד אלגוריתמים ועל תכונותיהם.

מתכון

הרי מתכון למוס שוקולד המובא מן הספר בישול צרפתי של סינקלייר ומלינובסקי (Weathervane Books, 1978, עמוד 73). החומרים, כלומר הקלטים, כוללים 8 אונקיות שוקולד חצי-מתוק שבור לחתיכות, 2 כפיות מים, $\frac{1}{4}$ כוס אבקת סוכר, 6 ביצים מופרדות, וכן הלאה. הפלטים הם בין 6 ל-8 קעריות של מוס שוקולד ערב לחך. הנה אפוא המתכון, או האלגוריתם:

ממיסים את השוקולד יחד עם שתי כפיות המים בסיר בעל תחתית כפולה. כשהשוקולד נמס, מערבבים לתוכו את אבקת הסוכר; מוסיפים בהדרגה את החמאה בחתיכות קטנות. מניחים בצד. מקציפים את החלמונים במשך 5 דקות בערך, עד שהם נעשים סמיכים וגונים לימון. יוצקים בעדינות את השוקולד על החלמונים ובוחשים יחד. מחממים שוב כדי להמיס את השוקולד, אם יש צורך. מוסיפים וניל ורום. מקציפים את החלבונים עד שהקצף יציב. מוסיפים תוך כדי ההקצפה 2 כפיות סוכר; ממשיכים להקציף עד שהקצף חזק. מקפלים בעדינות את החלבונים המוקצפים לתוך תערובת השוקולד והחלמונים. מוזגים לקעריות אישיות. מקררים במשך 4 שעות לפחות. מגישים עם קצפת, אם רוצים. הכמות מספיקה ל-6 עד 8 מנות.

זוהי ה"תוכנה" המתייחסת להכנת המוס; זהו האלגוריתם המפקח על תהליך הכנת המוס מן החומרים. התהליך עצמו מבוצע על-ידי ה"חומרה", שהיא במקרה זה האדם המכין את המוס והכלים השונים - הסיר, מכשיר החימום, המקצף, הכפיות, השעון וכיוצא באלה.

רמות פירוט

הבה נעמיק קצת יותר בהוראות היסודיות ביותר המצוינות במתכון זה. חשבו נא למשל על ההוראה: "מערבבים לתוכו את אבקת הסוכר". מדוע לא כתבו במתכון לוקחים מעט אבקת סוכר, שופכים אותה לתוך השוקולד הנמס, מערבבים, לוקחים עוד קצת אבקת סוכר, שופכים פנימה,

מערבבים...?" או מדוע לא כתוב בפירוט: "לוקחים 2365 גרגרי אבקת סוכר, שופכים אותם לתוך השוקולד המומס, לוקחים כפית ומערבבים את אבקת הסוכר בתנועות סיבוביות..."? ומדוע לא ביתר דיוק: "מזיזים את הזרוע לעבר החומרים בזווית של 14°, במהירות של בערך 46 ס"מ לשנייה..."? התשובה ברורה, כמובן. החומרה יודעת כיצד לערבב אבקת סוכר בתוך שוקולד מומס ואינה זקוקה לפירוט נוסף. אם כן, יש מקום לשאלה הבאה: האם מותר להניח שהחומרה כבר יודעת להכין תערובת שוקולד, אבקת סוכר וחמאה? במקרה כזה אפשר למחוק את כל החלק הראשון של המתכון ולכתוב בפשטות: "מכינים תערובת שוקולד". כמקרה קיצוני, אולי החומרה יודעת מאלה להכין מוס שוקולד. אזי אפשר להמיר את המתכון כולו בהוראה: "מכינים מוס שוקולד". אכן, בהינתן רמת מומחיות כזאת של החומרה, שורת הוראה אחת היא המתכון המושלם למוס שוקולד. מתכון אחרון וקצר זה הינו ברור, נטול שגיאות ומובטח לנו שיפיק את הפלטים הרצויים.

ניסויים מחשבתיים כאלה מבהירים את חשיבותה של רמת הפירוט בהצגת הוראות היסוד המופיעות באלגוריתם. הוראות אלה צריכות להתאים ליכולותיה המסויימות של החומרה ולרמת ההבנה של הקורא או של המשתמש הפוטנציאלי באלגוריתם.

הבה נתבונן בדוגמה נוספת המוכרת לנו כבר משלב מוקדם של חיינו, דוגמה מתחום החישוב - כפל שיטתי של מספרים. נניח שעלינו לכפול 528 ב-46. אנו יודעים כמובן כיצד לעשות זאת: כופלים 8 ב-6 ומקבלים 48; כותבים את ספרת היחידות של התוצאה - 8, וזוכרים את ספרת העשרות - 4; לאחר מכן כופלים 2 ב-6, מוסיפים את ה-4 ומקבלים 16; כעת כותבים את ספרת היחידות לשמאלה של הספרה 8 שכתבנו קודם, וזוכרים את ספרת העשרות - 1, וכן הלאה.

גם כאן ניתן לשאול את אותן שאלות. מדוע "כופלים 8 ב-6"? מדוע לא "מסתכלים על המשבצת הנמצאת בשורה השמינית ובעמודה השישית של לוח הכפל", ומדוע לא "מוסיפים 6 לעצמו 8 פעמים"? באותו אופן, מדוע איננו יכולים לפתור את הבעיה כולה בבת אחת באמצעות האלגוריתם הפשוט והמספק "כופלים את שני המספרים"? השאלה האחרונה עדינה למדי: מדוע מותר לנו לכפול 8 ב-6 באופן ישיר, אולם איננו מורשים לכפול ישירות את 528 ב-46? שוב, ברור שרמת הפירוט היא שתקבע במידה רבה, עד כמה מוכנים אנו לקבל את האלגוריתם המוצע כפתרון מוצלח לבעיה. בדוגמה זו מניחים שהחומרה הרלוונטית (במקרה זה אנו עצמנו) אינה יודעת לחשב ישירות כמה הם 528 פעמים 46, אך היא כן מסוגלת לחשב כמה הם 8 פעמים 6. למרות שבשני המקרים מדובר בפעולה הנראית לכאורה זהה.

דוגמאות אלה מצביעות על כך שיש צורך להסכים, מיד בהתחלה, מה הן פעולות היסוד שכל אלגוריתם רשאי להורות על ביצוען. מבלי לעשות זאת, אין טעם לנסות למצוא אלגוריתמים לשום דבר. יתרה מזאת, בעיות שונות קשורות באופן טבעי לסוגים שונים של פעולות יסוד. מתכונים עניינם בחישה, ערבוב, מזיגה וחימום; בכפל מספרים יש לחבר ולכפול ספרות וכן לזכור ספרה; כאשר מחפשים מספר טלפון בספר - מדפדפים, מזיזים אצבע לאורכה של רשימה ומשווים את השם הנתון לשם שהאצבע מצביעה עליו ברגע מסוים.

לגבי סוג האלגוריתמים המסוים שבו בכוונתנו לדון, יש צורך לציין בבהירות ובדיוק מה הן הוראות היסוד. איננו יכולים להרשות את קיומן של הוראות מעין "מקפצים את החלבונים עד שהקצף יציב", היות שקצף עשוי להצטייר באופן שונה לחלוטין לאנשים שונים! כמו כן, פעולות צריכות להיות מובחנות במידה מספקת ממשפטים שאינם מציינים פעולות, כגון "הכמות מספיקה ל-6 עד 8 מנות". לקטעים מעורפלים כגון "במשך 5 דקות בערך" אין מקום באלגוריתם שאותו אמור מחשב לבצע. הוא הדין למשפטים רב-משמעיים מעין "מגישים עם קצפת, אם רוצים" (האם ההגשה היא התלויה ברצונותיו של אדם, או הוספת הקצפת?). מתכונים למוס, בניגוד לאלגוריתמים שיעניינו אותנו כאן, מקבלים יותר מדי דברים כמובנים מאלהם, במיוחד את העובדה שאדם מהווה חלק מן החומרה. איננו יכולים להרשות לעצמנו מותרות כאלה, ולכן עלינו להיות תובעניים יותר כלפי האלגוריתמים. איכותו הכללית של האלגוריתם תלויה אפוא באופן קריטי בבחירת פעולות היסוד הקבילות ובמידת התאמתן לעניין הנדון.

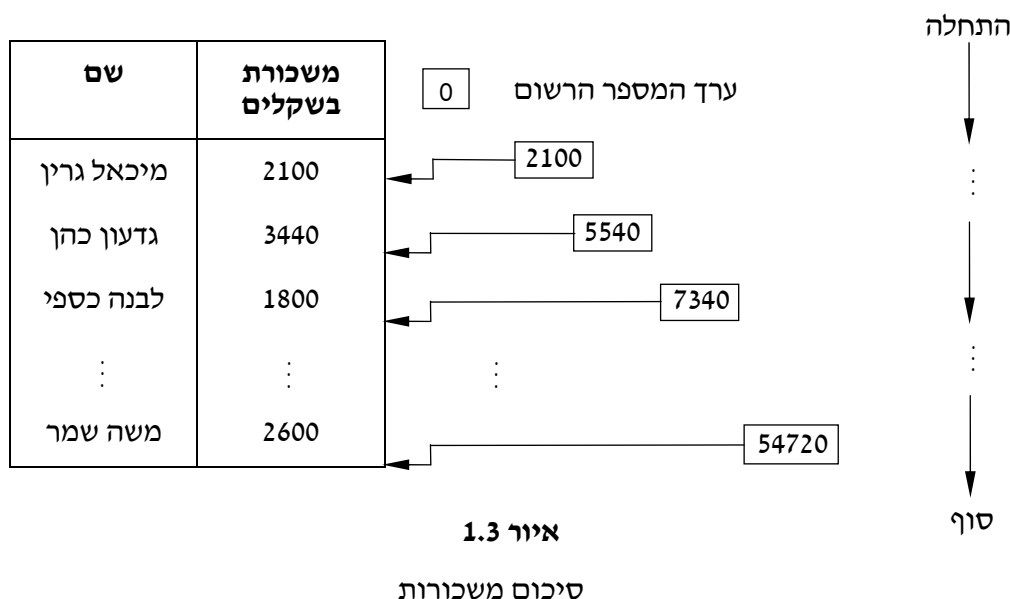
ייתכן שהנך תמה עתה על הניגוד שבין הדיון כאן לבין דיונונו לעיל על הסיביות והפעולות עליהן. שם הצהרנו שמחשבים יכולים לבצע למעשה אך ורק פעולות פשוטות ביותר על עצמים פשוטים ביותר, ואילו כאן אנו ממליצים להשתמש בפעולות יסוד מרמות פירוט שונות לתכנון אלגוריתמים שונים. ובכן, יש פה אכן ניגוד, אך זה ייפתר בפרק 3, כאשר נדון על אופן קשירת אלגוריתמים למחשבים ממשיים.

אלגוריתמים קצרים עבור תהליכים ארוכים

נניח שקיבלנו רשימה של רשומות כוח אדם, רשומה אחת עבור כל עובד בחברה מסוימת. כל רשומה מכילה את שם העובד, פרטיו האישיים ומשכורתו. אנו מעוניינים לסכם את משכורות כל העובדים ולהציג את הסכום שהתקבל. הנה אלגוריתמם לביצוע משימה זו:

1. רושמים את המספר 0;
2. סורקים את הרשימה, עובד אחר עובד, ובכל פעם מחברים אל המספר הרשום את ערך משכורתו של העובד;
3. כאשר מגיעים לסוף רשימת העובדים, מציגים את המספר הרשום כפלט.

ראשית, עלינו לשכנע את עצמנו שאלגוריתם פשוט זה אכן מבצע את הנדרש. המספר הרשום, זה שאפשר לזכור את ערכו או לרשמו על פיסת נייר, מקבל תחילה את הערך אפס. לאחר שמבצעים את החיבור בשלב (2) עבור העובד הראשון, מקבל המספר את ערך משכורתו של עובד זה. לאחר סריקת הרשומה השנייה, ערכו הוא סכום המשכורות של שני העובדים הראשונים, וכן הלאה. לבסוף ערכו הוא כמובן סכום כל המשכורות (ראה איור 1.3).



מעניין שטקסט האלגוריתם קצר ואורכו קבוע, אולם התהליך שהוא מתאר ושעליו הוא מפקח, משתנה לפי אורכה של רשימת העובדים ועשוי להיות ארוך מאוד מאוד. שתי חברות, אחת המעסיקה עובד אחד בלבד ושנייה המעסיקה מיליון עובדים, יכולות שתיהן להזין את האלגוריתם ברשימות העובדים שלהן, ובעיית סיכום המשכורות תיפתר עבור כל אחת מהן באופן משביע רצון. מובן שהתהליך לא יארך זמן רב עבור החברה הראשונה, אך ידרוש זמן ממושך למדי עבור השנייה. האלגוריתם, בכל מקרה, אינו משתנה.

לא זו בלבד שטקסט האלגוריתם קצר ואורכו קבוע, אלא שתי החברות, הקטנה והגדולה, דורשות כל אחת "מספר רשום" אחד בלבד לצורך ביצוע החישוב. משמע שמספר "כלי העזר" הדרושים אף הוא קטן וקבוע. כמובן, ערכו הסופי של המספר הרשום יהיה מן הסתם גדול יותר עבור חברות גדולות יותר, אולם תמיד יהיה זה מספר יחיד.

הבעיה האלגוריתמית

יש לנו אפוא אלגוריתם קבוע הגורם לביצועם של תהליכים רבים באורכים שונים, כאשר התהליך עצמו ומשך זמן ביצועו תלויים בקלטים לאלגוריתם. ואומנם, אפילו בדוגמה הפשוטה של סיכום המשכורות, קיים מגוון של קלטים אפשריים: חברות בנות עובד אחד, חברות המעסיקות מיליון עובדים, חברות שבהן חלק מן המשכורות הן בגובה אפס, או חברות שבהן כל המשכורות שוות. לעתים צריך אלגוריתם לעבוד על קלטים מוזרים, למשל חברות שאין בהן כלל עובדים, או חברות המעסיקות אנשים המקבלים שכר שלילי (כלומר, עובדים המשלמים לחברה עבור התענוג לעבוד בה).

למעשה, אנו מניחים שאלגוריתם המשכורות יפעל כראוי על מספר אינסופי של קלטים. עקרונית קיים מספר אינסופי של רשימות עובדים המתקבלות בהחלט על הדעת, ועל האלגוריתם להיות מסוגל לסכם את המשכורות בכל אחת מהן, אם זו תינתן לו כקלט.

עניין זה, מספר אינסופי של קלטים אפשריים, אינו משתלב היטב בהקבלה שבין אלגוריתם למתכון. למרות שמתכון מוס השוקולד אמור לעבוד באופן מושלם, ואין זה משנה כמה פעמים נשתמש בו, כמויות המרכיבים המופיעים בו קבועות בדרך כלל, ולכן למתכון יש למעשה קלט אפשרי אחד בלבד. אולם, אפשר להפוך את המתכון לכללי; רשימת החומרים תיראה אז בערך כך: X אונקיות שוקולד שבור לחתיכות, $\frac{X}{4}$ כפיות מים, $\frac{X}{32}$ כוסות אבקת סוכר...", והשורה אחרונה תהיה "הכמות

מספיקה ל- $\frac{3X}{4}$ עד X מנות". זה יתאים יותר לתפישה האמיתית של אלגוריתם. המתכון שהצגנוהו

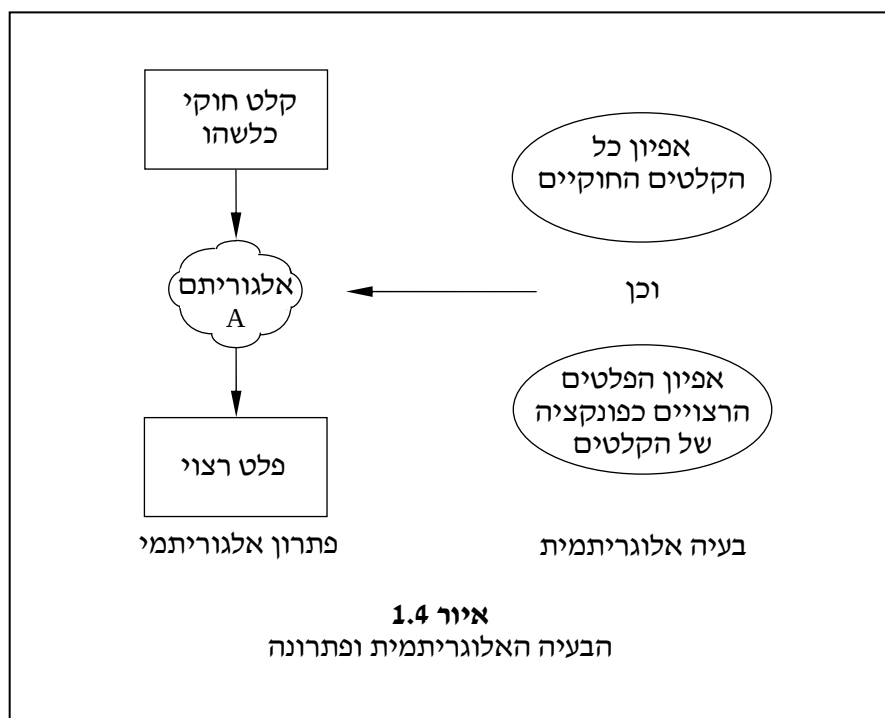
הוא אלגוריתם בעל אופי טריוויאלי משהו, היות שהוא "תפור" עבור קבוצה מסוימת בלבד של חומרים. אומנם אפשר לבצע אותו (או במינוח אלגוריתמי - להריץ אותו) מספר פעמים כרצוננו, אך תמיד עם אותו קלט, שהרי לצורך העניין, כוס אחת של קמח זהה לחלוטין לכל כוס אחרת של קמח.

על הקלט עצמו להיות חוקי ביחס למטרת האלגוריתם. מכאן אפשר ללמוד שרשימת רבי המכר של הניו-יורק-טיימס לא תהיה קבילה למשל כקלט של אלגוריתם סיכום המשכורות, ושחמאת בוטנים וריבה לא יתקבלו כחומרים אפשריים עבור מתכון המוס. דרישה זו מחייבת סוג מסוים של אפיון הקלטים המותרים. מישהו חייב אפוא להגדיר בדיוק אילו רשימות עובדים חוקיות ואילו אינן חוקיות; היכן בדיוק מופיעה המשכורת ברשימה; האם היא מופיעה בנוסח מלא (למשל 25000 שקלים לשנה) או שמא בנוסח מקוצר כלשהו (למשל 25K שקלים); היכן מסתיימת רשומה של עובד אחד ומתחילה רשומה של עובד אחר, וכן הלאה.

במונחים הכלליים ביותר, מתכונים, או אלגוריתמים, הינם פתרונות לסוג מסוים של בעיות הנקראות **בעיות אלגוריתמיות**, או **בעיות חישוביות**. בדוגמת המשכורות ניתן להציג את הבעיה כחיפוש אחר מספר המהווה את סכום כל המשכורות מתוך רשימת עובדים של חברה. אורך הרשימה יכול להשתנות, אולם עליה להיות מאורגנת תמיד בצורה מסוימת. במקרה זה ובמקרים דומים ניתן לומר שהבעיה היא לגלות את תוכנה של "קופסה שחורה" המאופיינת על-ידי הגדרה מדויקת הן של הקלטים החוקיים והן של הפלטים הנדרשים כפונקציה של קלטים אלה. בהגדרת הפלטים כוונתנו לדרך שבה כל פלט תלוי בקלט (ראה איור 1.4). בעיה אלגוריתמית נפתרת כאשר נמצא האלגוריתם המתאים. הקופסה השחורה מתמלאת אז תוכן; היא "עובדת" על פי האלגוריתם. במילים אחרות, הקופסה השחורה יכולה להפיק את הפלט המתאים מכל קלט חוקי על-ידי ביצוע התהליך שאותו משרה האלגוריתם ושעליו הוא מפקח. המילה "מכל" במשפט הקודם חשובה מאוד; איננו מעוניינים בפתרונות שאינן פועלים כיאות עבור כל הקלטים. נקל למצוא פתרון שעובד כראוי עבור חלק מן הקלטים החוקיים בלבד. כדוגמה קיצונית, נעיין באלגוריתם הטריוויאלי:

(1) הפק 0 כפלט.

אלגוריתם זה עובד באופן יוצא מן הכלל עבור כמה רשימות עובדים מעניינות: אלה שאין בהן עובדים כלל, אלה שבהן מרוויחים כל העובדים 0.00 ש"ח (או כפולות שלו), או אלה שבהן מקזזות בדיוק המשכורות החיוביות את השליליות.



בהמשך נטפל בנושאים הנוגעים ליעילותם של אלגוריתמים ולשאלת האפשרות לממשם הלכה למעשה. כאן אנו מציבים רק את הדרישה המינימלית, שאלגוריתם אכן יפתור את הבעיה, ולו גם בחוסר יעילות. מובן שהבעיה עצמה יכולה להנחות את התנהגות האלגוריתם גם בתגובה לקלטים לא רצויים, אך יש לשים לב שקלטים אלה, למרות שאינם רצויים, הם חוקיים. למשל, בעיית סיכום המשכורות יכולה בהחלט לכלול את הדרישה לצבור רשומה של עובד במאגר מיוחד, אם במקום שבו אמורה להירשם המשכורת לא רשום מספר אלא למשל מרכאות או נתונים חסרי מובן אחרים. מאגר זה יימסר אחר כך לטיפולו של המשרד האחראי על תשלום המשכורות. רשימה חריגה זו של עובדים היא בכל זאת חוקית, אלא שהטיפול בה אינו הטיפול הרגיל אלא מיוחד, המתאים לטבעה. סינון קלטים לא חוקיים הוא אפוא באחריותה של הבעיה האלגוריתמית, בעוד שהטיפול במחלקות מיוחדות לקלטים יוצאי דופן או בלתי רצויים הוא באחריותו של האלגוריתם עצמו.

אילוצים על פעולות יסוד

יש עוד עניין חשוב שעלינו להזכיר כעת, והוא נוגע לביצוען של פעולות היסוד בהתאם להנחיות האלגוריתם. כוונתנו לכך שעל ביצועה של כל אחת מפעולות אלה להסתיים תוך זמן סופי, שאם לא כן לעולם לא יסיים האלגוריתם את עבודתו. לכן, פעולות שמשך ביצוען אינסופי תידחינה על הסף. פעולות שאין חסם תחתון על משך זמן ביצוען אסורות גם הן, עובדה שאינה זקוקה כמעט להצדקה, שכן אין להעלות על הדעת שמכונה תוכל אי פעם לבצע פעולות תוך כדי משך זמן הולך וקטן. מהירות האור, למשל, תשמש תמיד כגבול עליון למהירותה של כל מכונה. יש להטיל אילוצים דומים על המשאבים (כלומר על כלי העבודה) המשמשים לביצוע פעולות היסוד, אולם לא נדון כאן בסיבות לכך.

אין ספק שהנחות אלה על פעולות היסוד אכן מתקיימות במחשבים ממשיים. פעולות היסוד לטיפול בסיביות, למשל, הן מדויקות וחד-משמעיות, וביצוען דורש כמויות מוגבלות של זמן ושל משאבים. מכאן, שאת התיאוריה האלגוריתמית שתואר להלן, ניתן ליישם ישירות לגבי בעיות המיועדות לפתרון באמצעות מחשב.

הבעיה ופתרונה: סיכום

לסיכום, בעיה אלגוריתמית כוללת:

1. אפיון האוסף, הסופי או האינסופי, של קבוצות הקלטים החוקיים האפשריים;

2. הגדרת הפלטים הרצויים כפונקציה של הקלטים.

אנחנו מניחים שתיאור פעולות היסוד המותרות, או תיאור קונפיגורציית החומרה יחד עם פעולות היסוד המיושמות בה, נתון מראש. פתרון לבעיה אלגוריתמית הוא אלגוריתם המורכב מהוראות יסוד אשר מבצעות פעולות מתוך קבוצת הפעולות שהוסכם עליה. אלגוריתם זה, לכשיוֹרץ על קבוצת קלט חוקי כלשהי, יפתור את הבעיה ויפיק את הפלט כנדרש. בפרקים 10 ו-11 נכליל במידה מסויימת רעיונות אלה, אולם עד אז נסתפק בהגדרה זו.

חשוב להכיר בקושי הרב להגיע לפתרון מספק של בעיה אלגוריתמית. בכך שפתחנו במתכון למוס שוקולד והמשכנו באלגוריתם הפשוט לסיכום משכורות, נוצר אולי הרושם שהדברים קלים. אין דבר הרחוק יותר מן האמת. בעיות אלגוריתמיות יכולות, למעשה, להיות קשות ומסובכות לפתרון, ומציאת פתרון מוצלח כרוכה לעתים בשנים של עבודה מפרכת. גרוע מכך: כפי שנראה בפרקים הבאים, לרבות מן הבעיות אין פתרון משביע רצון, ולאחרות אין פתרון כלל. מעמדן של בעיות רבות, בכל הנוגע לקיומן של פתרון אלגוריתמי טוב, אינו ידוע עדיין, וזאת למרות עמל רב שהושקע על-ידי מספר גדול של חוקרים.

מובן שלא נוכל להדגים את הנושאים שבהם מטפל ספר זה באמצעות דוגמאות ארוכות ומסובכות, אולם יש באפשרותנו לקבל מושג על הקושי בתכנון אלגוריתמים, אם נחשוב על הבעיות האלגוריתמיות הבאות (המובאות כאן בניסוח לא פורמלי): בבעיה אחת הקלט הוא מצב חוקי של לוח שחמט (כלומר תיאור המצב ברגע נתון של מהלך משחק שחמט), והפלט הוא המהלך הטוב יותר ללבן (כלומר תיאור המהלך שאם יבצע אותו הלבן, יהיו סיכויי לנצח במשחק הטובים ביותר). בעיה אחרת עוסקת בחלוקת עיתונים. נניח שיש לחלק 20000 עיתונים ל-1000 נקודות חלוקה ב-100 ערים שונות בעזרת 50 משאיות. הקלט כולל את המרחקים שבין הערים ואת המרחקים שבין מקומות שונים בכל עיר, את מספר העיתונים הנדרשים בכל מקום, את המיקום הנוכחי של כל משאית, את מספר העיתונים שיכולה כל משאית להוביל, וכן פרטים על הנהגים הזמינים, כולל מקום הימצאם הנוכחי. הפלט הוא רשימה המתאימה נהגים למשאיות, ומכילה מסלולי נסיעה מפורטים לכל אחת מן המשאיות, כך שמספר הקילומטרים הכולל שתיסענה כל המשאיות גם יחד יהיה מינימלי. למעשה, הבעיה דורשת אלגוריתם הפועל היטב עבור כל מספר של עיתונים, מקומות, ערים ומשאיות, כאשר מספרם של אלה גם הוא משתנה ומהווה חלק מן הקלט.

לפני שנוכל לדון בנושאים כגון נכונות ויעילות, או לטפל בשאלות עמוקות יותר הנוגעות לאופיים של פתרונות לבעיות אלגוריתמיות שונות או לעצם קיומם של פתרונות כאלה, עלינו ללמוד יותר על מבנה האלגוריתם ועל מבנה העצמים שבהם הוא מטפל.