אלגוריתמים - תרגיל 4

28 בינואר 2024

בתרגיל זה אתם מתבקשים לפתור בעיות שונות באמצעות תכנון דינמי. ודאו שהפתרון שלכם כולל:

1. ניסוח בהיר של תתי הבעיות
2. נוסחת רקורסיה והסבר לבנייתה
3. תיאור של הטבלה, סדר המילוי שלה ואופן חילוץ הפיתרון
4. ניתוח של זמן הריצה.

בשאלות שבהן לא הוגדרו דרישות סיבוכיות, זמן הריצה צריך להיות פולינומי בגודל הקלט.
אין צורך להוכיח את נכונות האלגוריתם.

שאלה 1

עד כה ראינו בעיות אשר ניתנות למילוי בטבלה חד מימדית או דו מימדית, כעת נראה שקיימות בעיות אשר דורשות טבלאות ממימדים גבוהים יותר.
נזכר בבעיית All Pairs Shortest Path שראינו בקורס מבני נתונים. הבעיה מוגדרת באופן הבא: בהנתן גרף ממושקל אנחנו מעוניינים למצוא את המרחק המינימלי בין כל שני קודקודים.

קלט: גרף מכוון $G = \langle V, E \rangle$ ופוקנציית משקל על הצלעות $w : E \to \mathbb{R}^+$. (נסמן: $n = |V|$).

פלט: מטריצה בגודל $n \times n$ כך שבתא i, j כתוב משקל מינימלי של מסלול מ-i ל-j.

כעת נציג 2 פתרונות דינמאיים שונים עבור הבעיה.
אנחנו נגדיר עבורכם את תתי הבעיות ונבקש מכם להמשיך את שאר הצעדים.
לשם נוחות, נניח כי הקודקודים ממוספרים בסדר $\{v_1, ..., v_n\}$. נשים לב כי מסלול אופטימלי לא יכלול מעגל. אם היה, היה ניתן להסירו ולקבל למסלול עם משקל יותר נמוך בסתירה לאופטימליות המסלול. לכן, מסלול בעל משקל מינימלי בין שני קדקודים יכיל לכל היותר $n - 1$ צלעות (במקרה הגרוע ביותר בו הוא עובר בין כל קודקודי הגרף).

- פתרון 1: אוסף תתי הבעיות: לכל $1 \leq i, j \leq n$ ו-$0 \leq m \leq n - 1$ נמצא את משקל המסלול המינימלי בין v_i ל-v_j המשתמש בלכל היותר m צלעות (נסמן מחיר זה ב-$f(i, j, m)$).
- פתרון 2: אלגוריתם Floyd Warshall
אוסף תתי הבעיות: לכל $1 \leq i, j \leq n$ ו-$0 \leq k \leq n$ נמצא את המשקל המינימלי של מסלול בין v_i ל-v_j שמשתמש רק בקודקודים $\{v_1, ..., v_k\}$ כקודקודי ביניים במסלול (נסמן מחיר זה ב-$g(i, j, k)$). כאשר $g(i, j, 0)$ אומר שהמסלול בין v_i ל-v_j לא משתמש בשום קודקוד ביניים.

נשים לב שקיימות תתי בעיות שאינן פתירות בהכרח, נגדיר את הערך שמוחזר במצבים אלו להיות ∞.

עבור פתרון 1 ופתרון 2, המשיכו את מילוי שאר הצעדים (הקפידו להתייחס לכל השלבים ובפרט למקרי הבסיס בנוסחת הרקורסיה, סדר מילוי הטבלה וניתוח זמן הריצה).
איזה פתרון דינאמי מבין 2 האפשרויות יעיל יותר מבחינת זמן ריצה?

שאלה 2

פלינדרום הוא מחרוזת $S = s_1 s_2 \dots s_n$ ששווה להיפוך שלה $S^R = s_n s_{n-1} \dots s_1$. למשל המחרוזות $X, ABCBA$ ו-QQ הן פלינדרומים אך המחרוזות XA, $BCDCA$ אינן.
בהינתן מחרוזת כקלט, הציעו אלגוריתם דינאמי שמחזיר תת-מחרוזת באורך מקסימאלי של הקלט שהינה פלינדרום.
למשל, בהינתן הקלט $ABCDB$, על האלגוריתם להחזיר את המחרוזת BCB (תת-מחרוזת לא חייבת להופיע ברצף בתוך הקלט). שימו לב כי השוואה בין מחרוזות באורך n מתבצעת ב-$O(n)$.

שאלה 3

יהי $G = (V, E)$ גרף מכוון בו כל צלע מתויגת באמצעות פונקציית תוויות $\sigma : E \to \Sigma$ (כאשר Σ מרחב תוויות סופי) ופונקציית משקל $w : E \to R^+$.
בהינתן קלט $S = s_1 s_2 \dots s_k$ המקיים $\forall 1 \leq i \leq k : s_i \in \Sigma$ וקודקוד התחלתי $v_0 \in V$, הציעו אלגוריתם דינאמי שמחזיר מסלול בעל משקל מקסימלי בגרף G המתחיל ב-v_0, כך שתיוג צלעות המסלול בידי שווה לS (כלומר, אם המסלול מורכב מהצלעות $e_1, e_2, e_3, \dots, e_k$, נדרוש כי $\forall i : \sigma(e_i) = s_i$). שימו לב כי ייתכן שלא קיים מסלול כזה. (רמז: זמן הריצה של האלגוריתם תלוי ב-k).

שאלה 4

נתונה מטריצה $M \in \{0,1\}^{n \times m}$, כלומר מטריצה שמימדיה $n \times m$ וכל איבריה הן 0 או 1. הציעו אלגוריתם אשר מוצא k מקסימלי עבורו קיימת תת-מטריצה בגודל $k \times k$ שכל איבריה הם 1. על האלגוריתם לרוץ בזמן $O(nm)$.
הגדרה: נאמר שמטריצה $A \in \{0,1\}^{k \times k}$ היא תת מטריצה רצפיה של M אם קיימים אינדקסים $1 \leq i \leq n-k+1$ ו $1 \leq j \leq m-k+1$ כך שמתקיים $A_{t,s} = M_{i+t-1, j+s-1}$ לכל $1 \leq t, s \leq k$.
לדוגמא, עבור הקלט הבא הפלט הוא 2:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

שאלה 5

חתול רוצה לחצות כביש סואן בעל n נתיבים במהירות ובבטחה. החתול מתחיל בנתיב 1 ומסיים בנתיב n, ובכל יחידת זמן יש לו 3 אפשרויות: 1) להישאר במקום, 2) להתקדם נתיב אחד קדימה, 3) לחזור נתיב אחד אחורה. יש N מכוניות, ולכל מכונית i ידוע כי היא נוסעת בנתיב l_i החל מזמן s_i ועד f_i, ובטווח זמן זה אסור לחתול להימצא באותו נתיב. פורמלית, אם נסמן ב-x_t את מיקום החתול בזמן t, אז

- הזמן נמדד ביחידות שלמות, כלומר $t \in \mathbb{N}$.
- בכל יחידת זמן t החתול נמצא באחד מהנתיבים, כלומר $x_t \in \{1, 2, \ldots, n\}$. הפעולות המותרות מכתיבות כי $x_{t+1} \in \{x_t, x_t + 1, x_t - 1\}$ לכל t.
- לכל מכונית i, ולכל זמן $s_i \leq t \leq f_i$, חייב להתקיים $x_t \neq l_i$ (אחרת החתול נדרס).

קלט:

- $n \in \mathbb{N}$: מספר הנתיבים,
- $N \in \mathbb{N}$: מספר המכוניות,
- $M \in \mathbb{N}$: הזמן המאוחר ביותר שבו מכונית עוברת,
- לכל $i = 1, \ldots, N$, נתונים l_i, s_i, f_i. ניתן להניח כי
  - $2 \leq l_i \leq n - 1$, מכוניות לא נוסעות בנתיבים 1 ו-n,
  - $0 \leq s_i < f_i \leq M$, הזמנים חוקיים וחסומים,
  - $s_i, f_i \in \mathbb{N}$, כל הזמנים נתונים ביחידת זמן שלמה.

פלט:

- הזמן המינימלי שייקח לחתול להגיע מנתיב 1 לנתיב n.

כתבו אלגוריתם תכנון דינמי אשר מוצא את הזמן הקצר ביותר בו יכול החתול לחצות את הכביש בלי להידרס. זמן הריצה של האלגוריתם צריך להיות $O\left(nNM\right)$ וסיבוכיות המקום $O\left(nM\right)$.