

תיאור של הפתרונות:

שאלה A:

בשאלה הזו ניסיתי להבין איך ליצור בצורה יעילה את כל התמורות האפשריות של אותיות המחרוזת, מבלי לחזור על אותה פרמוטציה פעמיים. הבנתי שאני יכולה להשתמש באלגוריתם DFS כדי לבנות את המחרוזות בצורה איטרטיבית – להתחיל ממחרוזת ריקה ומהמילה המקורית, ובכל שלב להוסיף אחת מהאותיות שעדיין לא השתמשתי בה. את האות שנוספה אני מסירה מהמילה המקורית, ומכניסה לסטאק את התחילית שנבנתה והאותיות שנותרו.

מכיוון ש-DFS מחזיר את הפרמוטציות מהסוף להתחלה, הבנתי שאם אמיין את האותיות של המילה המקורית מראש בסדר יורד, אז הפרמוטציות שיתקבלו יהיו ממוינות בסדר עולה. לכן, אפשר פשוט להדפיס כל פרמוטציה שמצאתי בלי למיין אותן בסוף.

בהתחלה שמרתי ב-visited רק את המילים השלמות שכבר נבנו (כלומר, כשלא נותרו אותיות להוסיף), אבל זמן הריצה לא היה מספיק טוב. כדי לייעל את זה, שמרתי ב-visited גם חישובים חלקיים: זוגות של התחילית שנבנתה והאותיות שנותרו. אם ראיתי כבר את אותו זוג, אין צורך להמשיך משם שוב. זה חסך הרבה חזרות מיותרות, והביא לפתרון שעומד בדרישות זמן הריצה.

שאלה B:

קיבלתי את הקלט ושמרתי את מצב הגינה הנוכחית במפה (map) ללא סדר, כדי שאוכל למצוא מפתח בזמן קבוע. במפה הזו המפתח הוא סוג הצמח, והערך הוא הכמות מכל סוג. אחרי כל פעולה, רציתי להדפיס את המצב הנוכחי של הגינה. כדי לבדוק אם הגינה הטרוגנית, מספיק לבדוק אם יש יותר מסוג אחד של צמח – כלומר, אם יש יותר ממפתח אחד במפה. בהתחלה ניסיתי לבדוק אם הגינה הומוגנית על ידי מעבר על כל המפתחות ובדיקה האם יש מפתח עם ערך גדול מ-1, אבל זה היה איטי מדי. כדי לשפר את הביצועים, שמרתי משתנה שסופר את כמות המפתחות שהערך שלהן גדול מ-1, ועדכנתי אותו בכל פעולה של הוספה או מחיקה. זה הוריד את זמן הריצה מסיבוכיות ריבועית לסיבוכיות זמן לינארית בגודל הקלט, ואחרי השינוי הזה הפתרון התקבל.

שאלה C:

כשקראתי את השאלה, חשבתי שבהנחה שכל הערכים במערך שונים – תמיד אפשר להכניס את הקערות אחת בתוך השנייה, ולכן מספר הקערות ה"נראות" הוא 1. כשיש כפילויות, הנחתי שבכל פעם שיש ערך שחוזר פעמיים, נצטרך קערה נוספת, אז הפתרון שלי היה: מספר הכפילויות + 1. אבל אז ראיתי שזה לא נותן תמיד את הפתרון המינימלי. בדוגמאות עם שתי כפילויות, ראיתי שאפשר להכניס גם כפילויות אחת לתוך השנייה – כלומר, לא חייבים קערה חדשה עבור כל כפילות. אז יצרתי וקטור של מפות, וכל פעם שעברתי על ערך במערך, בדקתי אם יש אחת מהמפות שכבר מכילה את הערך הזה. אם לא, הכנסתי את הערך לאותה מפה. אחרת, יצרתי מפה חדשה (שזה שקול לפתיחת קערה חדשה). כך, מספר הקערות ה"נראות" שווה לאורך הווקטור שקיבלתי בסוף.

שאלה D:

בהתחלה ניסיתי למיין את המילים לפי הסדר הלקסיקוגרפי הרגיל ($a < b$), אך הבנתי שזה לא תמיד עובד, במיוחד כשאחת המילים היא תחילית של השנייה. זה גרם לי להבין שמה שחשוב הוא לא סדר המילים בפני עצמו,

אלא השרשור שנוצר בעקבות הסדר. לכן, במקום להשוות בין a ל- b , השוויתי בין $a + b$ ל- $a + b$ (השרשור שלהן) ובחרתי את הסדר שמייצר תוצאה קטנה יותר. לאחר המיון לפי כלל זה, מיינתי את מילות הקלט ושרשרתי אותן אחת לשניה.

שאלה E:

הבנתי שכדי ליצור את החלוקה הטובה ביותר בין הנקודות, אני צריכה לבחור קו מפריד להעביר או בציר ה- x או בציר ה- y . לכן, מיינתי את הנקודות פעם אחת לפי x ופעם אחת לפי y . אחרי זה, עבור הנקודות הממוינות בדקתי מה הערך שהרווחנו ע"י חיסור של המלבן שנוצר מכל הנקודות מסכום השטחים של הריבועים שנוצרו מהחלוקה ל-2 של הנקודות. עשיתי שני מעברים כדי לחשב את המינימום והמקסימום של הנקודות שראינו עד כה בזמן לינארי בצורה נוחה.

שאלה F:

כתבתי מימוש של הפיתרון הנאיבי שרץ בזמן n^2 . עבור כל "נקודת התחלה" במערך חישבתי את כל תתי הקבוצות הרציפות שמתחילות מנקודה זו, ועבור כל איבר במערך חישבתי את ערך האי-יציבות באופן שתואר בשאלה. עדן הביאה לי רמז על שימוש במחסנית מונוטונית. קראתי על המבנה נתונים הזה וניסיתי לחזור על הדוגמא בתרגול אבל לא הבנתי איך להשתמש בו כדי לשפר את זמן הריצה של השאלה. ניסיתי לחשוב על סכום המקסימומים פחות סכום המינימומים כמו שדיברנו בשעת הקבלה אבל לא הצלחתי להתקדם משם.