Вказівки щодо розв'язання завдання № 19 відбірково-тренувальних зборів команди міста Києва

1. Кур’єри. Алгоритм подамо коментарями авторського розв’язання.

1. Зчитування значень N і M.
2. Зчитування початкових позицій кур'єрів.
3. Впорядкування позицій кур'єрів за зростанням координати.
4. Зчитування позиції всіх поточних замовлень, і для них:
    - перевірка крайніх кур'єрів (чи поточне замовлення лівіше найлівішого з кур’єрів або правіше найправішого з них);
    - визначення найближчого кур'єра двійковим (бінарним) пошуком;
    - збільшення пройденої відстані на відповідну величину й відповідна зміна координати одного кур'єра.

4. Виведення результату

Порядок розташування кур'єрів незмінний при виконанні всіх замовлень.

2. Робочий календар. Спочатку розглянемо більш просту задачу: в кожному тижні один день ($M = 1$). Тоді календар можна подати одновимірним масивом C довжиною N, а робоча зміна — це пара чисел, що стоять поряд. Така спрощена задача розв’язується методами динамічного програмування. Подамо один з методів розв’язання.

Запровадимо допоміжні масиви A і B**:**

- в елементі $A[i]$ будемо зберігати найбільше сумарне число сприятливості яке можна отримати, якщо всі робочі зміни відпрацьовані за перші i тижнів (днів), а останній день останньої робочої зміни — це день з номером i**;**
- в елементі $B[i]$ будемо зберігати найбільше сумарне число сприятливості за перші i днів, але за умови, що i-ий робочий день не належить ні до якої робочої зміни.

Справджуються такі висловлювання:

1. $B[1] = 0$, $A[1] = 0$,
2. Для всіх $i > 1$ $A[i] = B[i-1] + C[i-1] + C[i]$
3. Для всіх $i > 1$ $B[i] = \max(B[i-1], A[i-1])$

За допомогою співвідношень 2 і 3 можна обчислити $A[i]$ та $B[i]$ для будь-якого i. Відповіддю задачі буде число $K = \max(A[N], B[N])$. Складність роботи алгоритму $O(N)$.

Тепер розглянемо задачу при $M > 1$ і $N > 1$. У цьому разі робоча зміна це пара чисел, що можуть стояти як поряд, так і через $M - 1$ число одне від одного.

Запровадимо такі поняття:

- Для тижня робочого календаря кожен робочий день або належить якійсь робочій зміні, або ні. Тобто кожному робочому тижню можна поставити у відповідальність масив довжиною M, елементами якого є 0 або 1. Причому на i-ому місці стоїть 1, якщо i-ий день належить робочій зміні, *що завершується до кінця цього тижня,* або 0 у іншому випадку. Такий масив з 0 та 1 будемо називати станом робочого тижня. Станом i-го робочого дня будемо називати i-ий елемент стану тижня, до якого цей день належить.

- Розглянемо стани двох послідовних робочих тижнів. Вартістю переходу між цими станами, назвемо максимально можливе сумарне число сприятливості тих робочих змін, що починаються на першому тижні у день зі станом 0, або на другому тижні у день зі станом 1. В обох випадках робоча зміна обов'язково має закінчуватись на другому тижні в день зі станом 1.

Запровадимо допоміжний двомірний масив $A[1..N, 0..2^M-1]$: в елементі $A[i, j]$ будемо зберігати найбільше можливе число сприятливості для перших i тижнів, причому стан тижня i відповідає запису числа j у двійковій системі числення.

Справджуються такі висловлювання.

1. Для всіх станів j: $A[0, j] = 0$;
2. Для всіх $i > 0$, $0 \le j < 2^M$: $A[i, j] = \max\limits_{k\in\{0,1,\dots,2^M-1\}} (A[i-1, k] + F(i-1, k, i, j))$,

де F — вартість переходу між станом k тижня $i-1$ та станом j тижня i.

За допомогою цих правил 1–2 можна обчислити $A[i, j]$ послідовно для всіх i, j.

Відповіддю задачі буде число $\max\limits_{k\in\{0,1,\dots,2^M-1\}} A[N, k]$.

Складність алгоритму є $O(N\cdot 2^{2M})$.

Подане розв'язання задачі ілюструє підхід, відомий як „*динаміка по краю*".

<u>3. Острови.</u> Архіпелаг з мостами є лісом (графом без циклів). Шукана відповідь — це максимальна сума прибутків на кожній компоненті зв'язності графу (дереві). В кожній компоненті зв'язності ми можемо зафіксувати довільну вершину і назвати її *коренем компоненти*. Це дасть нам змогу *орієнтувати усі ребра* так, щоб вони ішли від кореня. Інакше кажучи, припишемо кожній вершині відстань від кореня, що є довжиною маршруту (у кількості ланок), який сполучає корінь і дану вершину. Далі орієнтуємо кожне ребро таким чином, щоб воно йшло від вершини з меншою відстанню до вершини з більшою відстанню. Таке орієнтування дає нам змогу розбити нашу задачу на підзадачі. Дійсно, розглянемо піддерево, утворене якоюсь вершиною разом з усім досяжними з неї вершинами, якщо рухатися у сторону «від кореня». Запровадимо такі позначення:

- `adj[v]` — множина суміжних з островом `v` островів;
- `income[v]` — дохід від шахт з острову `v`;
- `closeCost[e]` — ціна закриття мосту `e`;
- `profitUse[v]` — відповідь для підзадачі для дерева з коренем в вершині `v`, якщо на острові `v` будуть відбуватись видобувні роботи;
- `profitLeave[v]` — те ж саме, що і `profitUse[v]` за умови, що на острові `v` *не* будуть проходити видобувні роботи.

Справджуються такі рекурентні співвідношення:

$\mathtt{profitUse}[v] = \mathtt{income}[v] +$

$$+\sum_{u\in \mathtt{adj}[v]}\max\{\mathtt{profitLeave}[u],\mathtt{profitUse}[u]-\mathtt{closeCost}[(v,u)]\}$$

$$\mathtt{profitLeave}[v]=\sum_{u\in \mathtt{adj}[v]}\max\{\mathtt{profitUse}[u],\mathtt{profitLeave}[u]\}$$

Авторське розв’язання використовує рекурсивну процедуру.