Cvičenie 4

Úloha 4.1. Dva najbližšie

Je daná množina bodov v rovine. Nájdite dvojicu dvoch bodov, ktoré sú v najmenšej možnej vzájomnej vzdialenosti.

Náčrt riešenia 4.1.

Podrobné riešenie si môžete vypočuť v rámci nahranej prednášky: https://youtu.be/TtEF9W4qmq8?t=2381

Usporiadajme si body podľa x-ovej súradnice a rozdeľme ich na dve polovice. Rekurzívne vieme zistiť výsledok pre obe časti. V jednej časti bude najbližšia vzdialenosť dvoch bodov k_1, v druhej k_2. Označme si $k = \min(k_1, k_2)$.

Ostáva nám tieto dve riešenia spojiť. Jedna možnosť je, že odpoveďou je k, pretože dva najbližšie body sa nachádzali v jednej polovici. To však nepokrýva všetky možné kombinácie. Ktoré sme vynechali? Neskúšali sme tie dvojice, v ktorých sú zvolené body v rozdielnych častiach. Potrebovali by sme preto vyskúšať ešte tie.

Ak by sme však skúšali všetky možné dvojice, riešenie by bolo $O(n^2)$ a iba sme si to zbytočne skomplikovali. Skúsme teda okresať počet možností, ktoré potrebujeme skúšať. Predstavme si deliacu čiaru (prechádzajúca x-ovou súradnicou, ktorá body delila na polovicu). Je jasné, že ak je niektorý z bodov od tejto čiary vzdialený viac ako k, neoplatí sa ho skúšať, je totiž príliš ďaleko, z rekurzie sme dostali lepší odhad riešenia.

Ostávajú nám teda dve zredukované množiny bodov v blízkosti deliacej čiary. Prinajhoršom tam však opäť môžu byť všetky vrcholy a možností je priveľa. Vyberme si teda jeden konkrétny bod v ľavej polovici. Koľko bodov z pravej sa s ním oplatí skúšať? Je jasné, že všetky body, ktorých y-ová súradnica sa líši od nášho bodu o viac ako k sú irelevantné. Ostáva nám teda len obdĺžnik $k \times 2k$. Čo ak je v ňom stále veľa bodov?

To sa už stať nemôže, pretože tieto body musia byť od seba všetky vzdialené aspoň k, opäť by sme totiž z rekurzie dostali menšiu hodnotu. Pre každý bod z jednej časti existuje preto najviac 6 bodov, ktoré sa oplatí vyskúšať na tej pravej.

V riešení si teda už len body v týchto dvoch pásoch okolo deliacej čiary usporiadame podľa y-ovej súradnice a prejdeme pomocou dvoch bežcov. Pre každý bod vyskúšame konštatne veľa bodov, zložitosť bude teda lineárna. Celé riešenie má zložitosť $O(n \log^2 n)$ (rekurzia nám dá jeden logaritmus a triedenie bodov podľa y druhý), dá sa to však implementovať aj v čase $O(n \log n)$.

Úloha 4.2. Palindrómy

Pre daný reťazec nájdite jeho najdlhší súvislý podreťazec, ktorý je palindrómom.

Náčrt riešenia 4.2.

Palindróm je slovo, ktoré sa číta rovnako odpredu ako odzadu. Najľahšie palindróm skontrolujeme tak, že si zoberieme jeho "stredä ideme od neho naraz doprava aj doľava a kontrolujeme rovnakosť znakov v tej istej vzdialenosti od neho.

Najjednoduchšie riešenie problému teda robí presne toto. Postupne si určí každý možný stred a bude skúšať, ako veľký palindróm z neho vie vytvoriť – v cykle bude skúšať vzdialenejšie a vzdialenejšie pozície, kým nenájde písmená, ktoré nie sú rovnaké. Takéto riešenie má zložitosť $O(n^2)$.

Ako to však vylepšiť? Uvedomme si nasledovnú vlastnosť. Pre jeden konkrétny stred existuje viacero z neho vychádzajúcich palindrómov. Ak totiž existuje palindróm dĺžky x, aj ľubovoľný kratšie slovo s týmto stredom bude palindrómom. Toto však platí aj naopak. Ak slovo dĺžky x už nie je palindróm, ani žiadne dlhšie s tým istým stredom palindrómom nebude, to kratšie totiž obsahuje ako podslovo.

Takúto vlastnosť, monotónnosť, využívame často pri binárnom vyhľadávaní. Pre zadaný stred by sme sa mohli pýtať: Existuje palindróm dĺžky x s týmto stredom? Ak je odpoveď áno, hľadáme dlhšie slová, ak nie, hľadáme kratšie.

Ako však overiť, či je slovo dĺžky x palindróm? Lineárne porovnanie by totiž bolo príliš pomalé, mali by sme ešte horšie $O(n^2 \log n)$ riešenie. Uvedomme si, že to čo hľadáme je vlastnosť rovnosť slov, akurát na jednej strany je to slovo napísané opačne. Pomôžeme si preto rolling hashom. Pre každý prefix zistíme jeho rolling hash. Toto vieme vypočítať v čase $O(n)$. Následne hash ľubovoľného podslova vieme zistiť ako rozdiel dvoch prefixových hashov – od hashu konca odčítame hash začiatku, čím presne odstránime všetko prebytočné.

A ak toto isté spravíme aj pre každý sufix, vieme zistiť hash ľubovoľného podslova v obrátenom smere. Pre daný stred palindrómu a jeho dĺžku x potom vieme rýchlo porovnať obe strany palindrómu. Celé riešenie má zložitosť $O(n \log n)$ – vypočíme prefixové a sufixové hashe a pre každý stred binárne vyhľadáme najdlhší palindróm.

Existuje aj deterministické riešenie tohto problému v čase $O(n)$ – Manacherov algoritmus https://en.wikipedia.org/wiki/Longest_palindromic_substring#Manacher's_algorithm.

Úloha 4.3. Dvojrozmerný pattern search

Je daná matica P („pattern“) a matica T („text“). Nájdite všetky výskyty matice P v matici T, ak viete, že jednotlivé riadky matice P sú navzájom rôzne. Za *výskyt* matice P v matici T považujeme vhodný počet za sebou idúcich riadkov a stĺpcov v matici T, ktoré obsahujú prvky matice P na príslušných pozíciách.

Náčrt riešenia 4.3.

Predstavme si, že pre každé políčko matice T vieme, či tam začína niektorý z riadkov matice P, teda buď tam máme číslo 0 (tu žiaden riadok nezačína) alebo číslo od 1 po p_r (počet riadkov matice P). V okamihu ako sa v niektorom stĺpci nachádzajú postupne čísla 1 až p_r po sebe, máme hľadaný výskyt. A overiť niečo také vieme jedným prechodom matice.

Ostáva teda zistiť, kde začína ktorý riadok. A to nie je nič iné ako hľadanie viacerých vzoriek v texte. Všetky riadky teda nahodíme do Aho-Corasickovej algoritmu a prejdeme pomocou neho riadky matice T.

Môžete sa zamyslieť, ako by sa riešila táto úloha, ak by riadky nemuseli byť všetky rôzne. Základ ostane ten istý, akurát treba upraviť kontrolu toho, či sú v stĺpcoch správne čísla.

Úloha 4.4. Ostreľovanie

Budeme m krát strieľať na upravený terč, ktorý tvorí (namiesto sústredných kružníc) nekonečne veľa navzájom podobných do seba vložených konvexných n-uholníkov obsahujúcich bod $O = [0, 0]$.

Najvnútornejší mnohouholník je daný (v cyklickom poradí) ako pole dvojíc súradníc $[x_1, y_1]$, $[x_2, y_2], \ldots, [x_n, y_n]$. Každý ďalší je obrazom prvého v rovnoľahlosti so stredom v bode O a koeficientom $2, 3, 4$, atď.

Za zásah vnútri prvého mnohouholníka dostaneme 0 trestných bodov. Za zásah, ktorý je vnútri $(k + 1)$-teho ale nie je vnútri k-teho, dostaneme k trestných bodov.

Na vstupe sú čísla n a m; potom nasleduje n usporiadaných dvojíc súradníc vrcholov prvého mnohouholníka, a m usporiadaných dvojíc súradníc zásahov. Môžete predpokladať, že žiaden zásah sa nenachádza presne na okraji niektorého mnohouholníka. Úlohou je vypočítať celkový počet trestných bodov.

Náčrt riešenia 4.4.

Zoberme stred O a nakreslime si z neho polpriamky vedúce cez všetky vrcholy nášho mnohouholníka. Celá rovina sa nám rozdelí do niekoľkých trojuholníkov. Hľadanie toho, kde padol každý zo zásahov rozdelíme do dvoch častí. V prvej zistíme, v ktorom z týchto trojuholníkov sa bod nachádza a v druhom, v ktorej úrovni (teda koľko bodov) je.

V okamihu ako vieme trojuholník vieme zistiť úroveň pomerne jednoducho cez podobnosť trojuholníkov. Ako teda zistiť výsek, v ktorom je bod? Na to vieme použiť binárne vyhľadávanie nad uhlom. Vyberieme si trojuholník a zistíme, či je zásah napravo alebo naľavo od neho. Trochu mätúce môže byť, že je to vlastne kruhové, ale jediné čo stačí je tento kruh "rozseknúť"na ľubovoľnom mieste.