KMI/ALGO3 — Algoritmy 3

5. Rozděl a panuj II: chytré rozklady

Jan Konečný

Katedra informatiky, PřF UP Olomouc

Kde jsme: sylabus KMI/ALGO3

- **Co znamená navrhnout algoritmus** — specifikace, invariant, složitost
- **Hladové algoritmy** — kdy je lokální volba bezpečná
- **Hladové algoritmy na grafech** — kostry, nejkratší cesty
- **Rozděl a panuj** — rekurence, Master theorem
- **Rozděl a panuj II** — chytré rozklady
- **Dynamické programování I** — hledání stavů
- **Dynamické programování II** — vícerozměrné stavy
- **Toky v sítích** — max-flow/min-cut jako návrhový nástroj
- **Systematické prohledávání** — hrubá síla a backtracking
- **Branch-and-bound** — meze přesného řešení, P a NP
- **Adversariální návrh** — minimax, alfa-beta
- **Když přesné optimum není realistické** — aproximace, heuristiky
- **Randomizace a syntéza** — jak si vybrat metodu

Dnešní téma

Minule jsme dělili vstup na poloviny a soustředili se na důkazy a rekurence. Dnes se budeme ptát navíc, zda lze změnit reprezentaci nebo vracenou informaci tak, abychom potřebovali méně rekurzivních volání nebo levnější spojení.

- Karatsuba nahradí jedno násobení několika sčítáními.
- Lineární výběr investuje práci do dobrého pivota, aby mohl zahodit konstantní část vstupu.
- Nejbližší dvojice bodů při skládání vynechá téměř všechny dvojice přes hranici.

Ve všech třech případech rozhoduje přesnější porozumění tomu, *co opravdu musíme spočítat*.

Společná myšlenka

Ve všech dnešních algoritmech nahrazujeme drahou operaci větším počtem levných operací, případně pomocnou informací. Asymptotické zlepšení z toho ovšem vznikne jen tehdy, když se tato výměna opakuje na všech úrovních rekurze — jednorázová úspora se v celkové složitosti ztratí.

Budeme používat nástroje z minulé přednášky:

- Master theorem,
- rekurzivní strom,
- substituci (a její pasti).

Karatsubovo násobení: přímočarý pokus

Dvě nezáporná celá čísla A, B, každé nejvýše n-bitové. Školní násobení potřebuje $\Theta(n^2)$ bitových operací.

Nechť n je sudé, $m = n/2$:

$$A = A_1 2^m + A_0, \qquad B = B_1 2^m + B_0.$$

Roznásobením

$$AB = A_1 B_1 \, 2^{2m} + (A_1 B_0 + A_0 B_1) \, 2^m + A_0 B_0.$$

Čtyři součiny polovičních čísel, sčítání a posuny za $\Theta(n)$:

$$T(n) = 4\,T(n/2) + \Theta(n) = \Theta(n^2).$$

Použili jsme tedy rozděl a panuj a nezrychlili vůbec nic.

Tři násobení místo čtyř

Smíšený člen není nutné počítat dvěma násobeními:

$$(A_1 + A_0)(B_1 + B_0) - A_1B_1 - A_0B_0 = A_1B_0 + A_0B_1.$$

Rekurzivně tedy spočítáme jen

$$z_2 = A_1B_1, \qquad z_0 = A_0B_0, \qquad z_1 = (A_1+A_0)(B_1+B_0) - z_2 - z_0$$

a složíme

$$AB = z_2\,2^{2m} + z_1\,2^m + z_0.$$

Drobnost: součty A_1+A_0 a B_1+B_0 mohou mít $m+1$ bitů, takže třetí volání násobí čísla o bit delší. Na asymptotiku to nemá vliv — jeden bit se schová do lineární práce se sčítáním.

Korektnost a příklad

Věta 1 (Korektnost)

Karatsubův algoritmus vrátí přesný součin vstupních celých čísel.

Důkaz.

Indukcí podle počtu bitů; základ indukce používá korektní přímé násobení. Rekurzivní volání správně vypočtou tři menší součiny, definice z_1 a distributivita dávají $z_1 = A_1B_0 + A_0B_1$, a dosazení do skládacího vzorce je přesně roznásobením $(A_1 2^m + A_0)(B_1 2^m + B_0)$.

Příklad 2 (Táž identita v desítkové soustavě)

$A = 1234$, $B = 5678$; $A_1{=}12, A_0{=}34, B_1{=}56, B_0{=}78$.

$$z_2 = 672, \quad z_0 = 2652, \quad z_1 = 46 \cdot 134 - 672 - 2652 = 2840$$

$$1234 \cdot 5678 = 672 \cdot 10^4 + 2840 \cdot 10^2 + 2652 = 7\,006\,652.$$

Proč jedno volání rozhoduje

$$T(n) = 3T(n/2) + \Theta(n) \quad \Longrightarrow \quad T(n) = \Theta\left(n^{\log_2 3}\right), \quad \log_2 3 \approx 1{,}585.$$

Rozdíl mezi čtyřmi a třemi voláními se na první úrovni zdá malý. V hloubce $\log_2 n$ je však počet listů místo

$$4^{\log_2 n} = n^2 \qquad \text{pouze} \qquad 3^{\log_2 n} = n^{\log_2 3}.$$

Úspora se tedy násobí na každé úrovni.

Model výpočtu

Počítáme bitové operace; sčítání a posun stojí $\Theta(n)$. V modelu, který považuje násobení libovolně velkých čísel za jednu operaci, by otázka neměla smysl — složitost vždy závisí na tom, které operace účtujeme.

A co praxe

- Rekurze se ukončuje dřív než u jednoho bitu — pro malé vstupy má školní metoda *nižší režii*.
- Karatsuba se vyplatí až nad určitou mezí, kterou ovlivňuje reprezentace čísel i konkrétní počítač.



A. Karatsuba, Yu. Ofman.
Multiplication of Multidigit Numbers on Automata.
Soviet Physics Doklady 7 (1963), 595–596.

Historicky šlo o první publikovaný algoritmus, který překonal kvadratickou hranici školního násobení.

Jak Karatsubův algoritmus vznikl

Roku 1960 vedl v Moskvě seminář Andrej Kolmogorov a vyslovil domněnku:

násobení n-ciferných čísel vyžaduje $\Omega(n^2)$ *operací.*

Během následujícího týdne našel třiadvacetiletý student Anatolij Karatsuba popsaný postup se třemi součiny — a domněnku vyvrátil.

Kolmogorov výsledek nadšeně přednášel na dalších seminářích a sepsal jej; Karatsuba se prý o vlastní publikaci dozvěděl až po jejím vydání.

Z historky plyne i praktické poučení: domněnka „rychleji to nejde“ bývá založena jen na tom, že nás nenapadl jiný rozklad problému.

Karatsuba binárně, celý výpočet

$A = 1011_2 = 11$, $B = 1101_2 = 13$; $n = 4$, $m = 2$.

$$A_1 = 10_2 = 2, \quad A_0 = 11_2 = 3, \qquad B_1 = 11_2 = 3, \quad B_0 = 01_2 = 1$$

	rekurzivní součin	hodnota
z_2	$A_1 B_1 = 2 \cdot 3$	6
z_0	$A_0 B_0 = 3 \cdot 1$	3
z_1	$(A_1+A_0)(B_1+B_0) - z_2 - z_0 = 5 \cdot 4 - 6 - 3$	11

$$AB = z_2 \cdot 2^4 + z_1 \cdot 2^2 + z_0 = 6 \cdot 16 + 11 \cdot 4 + 3 = 96 + 44 + 3 = 143$$

a skutečně $11 \cdot 13 = 143$.

Ve třetím řádku má činitel $A_1+A_0 = 5$ *tři* bity, tedy o jeden víc než $m = 2$. To je drobnost, kterou jsme schovali do lineární práce se sčítáním.

Kam se dá jít dál

metoda	rok	čas násobení n-bitových čísel
školní	—	$\Theta(n^2)$
Karatsuba	1962	$\Theta(n^{1{,}585})$ (3 součiny ze 2 částí)
Toom–Cook (k částí)	1963	$\Theta(n^{\log_k(2k-1)})$
Schönhage–Strassen	1971	$O(n \log n \log\log n)$ (přes FFT)
Harvey–van der Hoeven	2019	$O(n \log n)$

- Toom–Cook je přímé zobecnění Karatsuby: rozděl na k částí a zjisti, že stačí $2k-1$ součinů (je to interpolace polynomu v $2k-1$ bodech).
- Poslední řádek je domnělé optimum — dolní mez $\Omega(n \log n)$ je otevřený problém.
- V praxi knihovny (GMP) mezi metodami *přepínají* podle velikosti čísla: školní → Karatsuba → Toom → FFT.

Táž myšlenka jinde: Strassenovo násobení matic

Součin dvou matic $n \times n$ blokově:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Přímočaře je to osm součinů matic $\frac{n}{2} \times \frac{n}{2}$ a lineární sčítání:

$$T(n) = 8\,T(n/2) + \Theta(n^2) = \Theta(n^3).$$

Táž myšlenka jinde: Strassenovo násobení matic

Součin dvou matic $n \times n$ blokově:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Přímočaře je to osm součinů matic $\frac{n}{2} \times \frac{n}{2}$ a lineární sčítání:

$$T(n) = 8T(n/2) + \Theta(n^2) = \Theta(n^3).$$

Strassen (1969) našel kombinaci, která vystačí se *sedmi* součiny (za cenu více sčítání):

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta\left(n^{\log_2 7}\right) = \Theta(n^{2{,}807\ldots}).$$

Je to táž výměna jako u Karatsuby: jedno drahé násobení za několik levných sčítání.

Výběr k-tého prvku

Definice 3 (Problém výběru)

Pole A s n navzájem různými prvky a $1 \leq k \leq n$. k-tým nejmenším prvkem rozumíme prvek, který by po seřazení ležel na pozici k. Speciálním případem $k = \lceil n/2 \rceil$ je medián.

Seřadit celé pole a vrátit k-tý prvek stojí $O(n \log n)$.

Potřebujeme ale znát celé pořadí, když nás zajímá jediná pozice?

Rozdělení kolem pivota

Vybereme pivot p a jedním průchodem rozdělíme prvky na

$$L = \{x : x < p\}, \qquad \{p\}, \qquad R = \{x : x > p\}.$$

- $|L| = k - 1$: pivot *je* odpověď;
- $k \leq |L|$: pokračujeme jen v L;
- jinak hledáme prvek pořadí $k - |L| - 1$ v R.

Quickselect

Vyber pivot, rozděl pole a rekurzivně prohledej *pouze* tu část, která obsahuje hledané pořadí.

Na rozdíl od quicksortu nikdy neřešíme obě části.

Rozdělení kolem pivota

Vybereme pivot p a jedním průchodem rozdělíme prvky na

$$L = \{x : x < p\}, \qquad \{p\}, \qquad R = \{x : x > p\}.$$

- $|L| = k - 1$: pivot *je* odpověď;
- $k \leq |L|$: pokračujeme jen v L;
- jinak hledáme prvek pořadí $k - |L| - 1$ v R.

Quickselect

Vyber pivot, rozděl pole a rekurzivně prohledej *pouze* tu část, která obsahuje hledané pořadí.

Na rozdíl od quicksortu nikdy neřešíme obě části.

$$\text{dobrý pivot:} \quad T(n) \leq T(3n/4) + \Theta(n) = \Theta(n)$$

$$\text{nejhorší případ:} \quad T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$$

Náhodný pivot má *očekávaný* lineární čas. Lze dobrý pivot zaručit *deterministicky*?

Quickselect na konkrétních číslech

Hledáme *medián* ($k = 5$) v poli $(7, 2, 9, 4, 1, 8, 3, 6, 5)$, $n = 9$; pivoty volíme libovolně.

pivot	rozdělení	pořadí pivota	pokračujeme
6	$L = \{2, 4, 1, 3, 5\}$, $R = \{7, 9, 8\}$	6.	$k = 5 \leq \lvert L\rvert$: v L
4	$L = \{2, 1, 3\}$, $R = \{5\}$	4.	$k = 5 > 4$: v R, $k' = 1$
5	$L = \emptyset$, $R = \emptyset$	1.	hotovo: 5

- Ve druhém řádku hledáme v R prvek pořadí $k - \lvert L\rvert - 1 = 5 - 3 - 1 = 1$.
- Celkem jsme prošli $9 + 5 + 1 = 15$ prvků místo $9 \log_2 9 \approx 29$ u úplného seřazení.
- Nešťastná volba pivota (třeba vždy nejmenší prvek) by dala $9 + 8 + 7 + \cdots = \Theta(n^2)$.

V praxi se proto pivot volí náhodně — očekávaně lineární čas za cenu, že záruka platí jen „v průměru“. Přesnou analýzu uděláme v přednášce 13.

Medián mediánů (BFPRT)

1. Rozdělíme prvky do skupin po pěti; poslední může být kratší.
2. Každou skupinu seřadíme a vezmeme její medián.
3. *Rekurzivně* nalezneme medián získaných mediánů a použijeme jej jako pivot.

Proč právě pět?

- Konstantní velikost skupiny $\Rightarrow$ medián skupiny za konstantní práci.
- Skupina je dost velká, aby pivot zaručeně oddělil *významnou část* vstupu.



M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, R. E. Tarjan.
Time Bounds for Selection.
JCSS 7(4) (1973), 448–461.

Jeden běh na patnácti prvcích

Příklad 4

Hledáme prvek pořadí $k = 11$ v poli

$$(12, 3, 47, 8, 29,\ \ 5, 41, 17, 33, 2,\ \ 26, 9, 38, 14, 50).$$

Mediány pětic po seřazení: 12, 17, 26. Medián mediánů: $p = 17$.

Rozdělení: $L = \{12, 3, 8, 5, 2, 9, 14\}$ (7 prvků),
$R = \{47, 29, 41, 33, 26, 38, 50\}$ — pivot je *osmý* nejmenší.

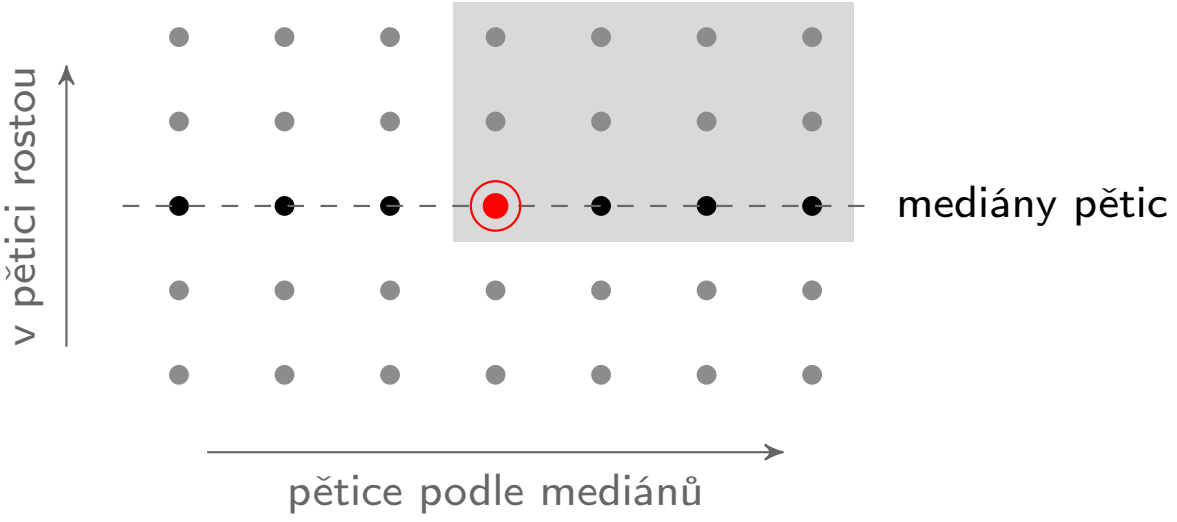
Protože $k = 11 > 8$, hledáme rekurzivně prvek pořadí $11 - 8 = 3$ v R a dostaneme 33.

(Kdybychom hledali $k = 8$, odpovědí by byl přímo pivot.)

Kolik prvků pivot zahodí

Lemma 5

Medián mediánů není menší než alespoň $3n/10 - O(1)$ *prvků a není větší než alespoň* $3n/10 - O(1)$ *prvků.*



Alespoň *polovina* skupinových mediánů není menší než pivot, a v každé takové pětici jsou alespoň *tři* prvky $\geq$ pivot. Skupin je $\approx n/5$, tedy $3 \cdot \frac{1}{2} \cdot \frac{n}{5} = \frac{3n}{10}$ prvků.

Rekurence, která není ve tvaru Master theoremu

Prvků menších než pivot je nejvýše $7n/10 + 6$ (a symetricky větších). Pivot sám hledáme mezi $\lceil n/5 \rceil$ mediány. S lineární prací dn mimo rekurzi:

$$T(n) \leq T\left(\left\lceil \tfrac{n}{5} \right\rceil\right) + T\left(\tfrac{7n}{10} + 6\right) + dn.$$

Pozorování

Součet velikostí rekurzivních vstupů je nejvýše

$$\frac{n}{5} + \frac{7n}{10} + O(1) = \frac{9n}{10} + O(1) \; < \; n.$$

Této rezervy důkaz využije.

Lineární deterministický výběr

Věta 6

Medián mediánů umožňuje nalézt k-tý nejmenší prvek v nejhorším případě v čase $\Theta(n)$.

Důkaz.

Dolní mez $\Omega(n)$ plyne už pro minimum: každý prvek kromě minima musí alespoň jednou prohrát porovnání.

Horní mez substitucí: $T(n) \leq cn$. Zvolíme c tak, aby $c \geq 20d$ a zároveň $T(n) \leq cn$ platilo pro všechna $n < 140$ (konečně mnoho vstupů). Pro $n \geq 140$ (pak oba vstupy $< n$):

$$T(n) \leq \frac{cn}{5} + c + \frac{7cn}{10} + 6c + dn = \frac{9cn}{10} + 7c + dn.$$

Z $c \geq 20d$ plyne $dn \leq cn/20$, z $n \geq 140$ plyne $7c \leq cn/20$, celkem $7c + dn \leq cn/10$, tedy $T(n) \leq cn$.

Konstantní posuny schované v $O(1)$ pohltila rezerva $cn/10$ mezi $9cn/10$ a cílem cn.

Proč se drahý pivot vyplatí

Medián mediánů sám používá rekurzi, což může působit jako zbytečná práce. Zaručené odstranění alespoň zhruba tří desetin prvků však způsobí, že součet práce všech úrovní tvoří klesající geometrickou řadu. Investujeme tedy lineární práci navíc, abychom spolehlivě zmenšili budoucí problém.

Srovnejte: quickselect s pivotem, který vždy dělí v poměru nejvýše $1 : 9$, splňuje $T(n) \leq T(9n/10) + cn$, a rozvinutí dává

$$cn + c\tfrac{9n}{10} + c\left(\tfrac{9}{10}\right)^2 n + \cdots \leq 10cn.$$

Používá se medián mediánů doopravdy?

- Konstanta ukrytá v $\Theta(n)$ je velká — řazení pětic, rekurzivní volání na $n/5$ prvcích, dvojí průchod pamětí.
- Náhodný pivot má očekávaně lineární počet porovnání (pro medián zhruba $3{,}4n$) a téměř nulovou režii. Pro reálná data proto vyhrává skoro vždy.

Introselect — kompromis, který se opravdu implementuje

Začni s náhodným pivotem. Sleduj, jak rychle se problém zmenšuje; klesne-li příliš pomalu, *přepni* na medián mediánů.

Dostaneme tak rychlost náhody i záruku $O(n)$ v nejhorším případě.

Je to táž myšlenka jako introsort z minulé přednášky: rychlá metoda s kvadratickým rizikem, jištěná pomalejší metodou se zárukou. Teoretický algoritmus tedy v knihovně nemusí být hlavní metodou, slouží ale jako pojistka.

Proč zrovna pětice? Zkusme trojice

Se skupinami po *třech* pokryje pivot v polovině skupin dva prvky místo tří, takže zaručeně odstraní jen zhruba

$$2 \cdot \frac{1}{2} \cdot \frac{n}{3} = \frac{n}{3}$$

prvků. Rekurence je

$$T(n) \leq T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + dn.$$

Součet velikostí rekurzivních vstupů je ale

$$\frac{n}{3} + \frac{2n}{3} = n,$$

tedy *přesně* n, ne méně. Substituce $T(n) \leq cn$ dá $T(n) \leq cn + dn > cn$ — krok neprojde.

Proč zrovna pětice? Zkusme trojice

Se skupinami po *třech* pokryje pivot v polovině skupin dva prvky místo tří, takže zaručeně odstraní jen zhruba

$$2 \cdot \tfrac{1}{2} \cdot \tfrac{n}{3} = \tfrac{n}{3}$$

prvků. Rekurence je

$$T(n) \leq T\left(\tfrac{n}{3}\right) + T\left(\tfrac{2n}{3}\right) + dn.$$

Součet velikostí rekurzivních vstupů je ale

$$\tfrac{n}{3} + \tfrac{2n}{3} = n,$$

tedy *přesně* n, ne méně. Substituce $T(n) \leq cn$ dá $T(n) \leq cn + dn > cn$ — krok neprojde.

Tato rekurence má řešení $\Theta(n \log n)$: s trojicemi tento postup lineární čas *nezaručí*.

Rezerva mezi $\frac{9n}{10}$ a n je tedy pro důkaz nezbytná.

Nejbližší dvojice bodů

Definice 7

Je dána množina $n \geq 2$ bodů v rovině. Úkolem je najít dva různé body s nejmenší eukleidovskou vzdáleností.

Přímé porovnání všech dvojic stojí $\Theta(n^2)$. Dělení podle x-ové souřadnice nabízí dva podproblémy — ale nejbližší dvojice může mít jeden bod v každé polovině.

Body seřadíme podle x, svislou přímkou rozdělíme, rekurzivně získáme δ_L, δ_R a položíme

$$\delta = \min\{\delta_L, \delta_R\}.$$

Zbývá zjistit, zda přes dělicí přímku vede dvojice bližší než δ.

Pás kolem hranice

Každý bod příčné dvojice bližší než δ musí ležet ve vzdálenosti $< \delta$ od dělicí přímky — stačí uvažovat pás šířky 2δ a body v něm seřadit podle y.

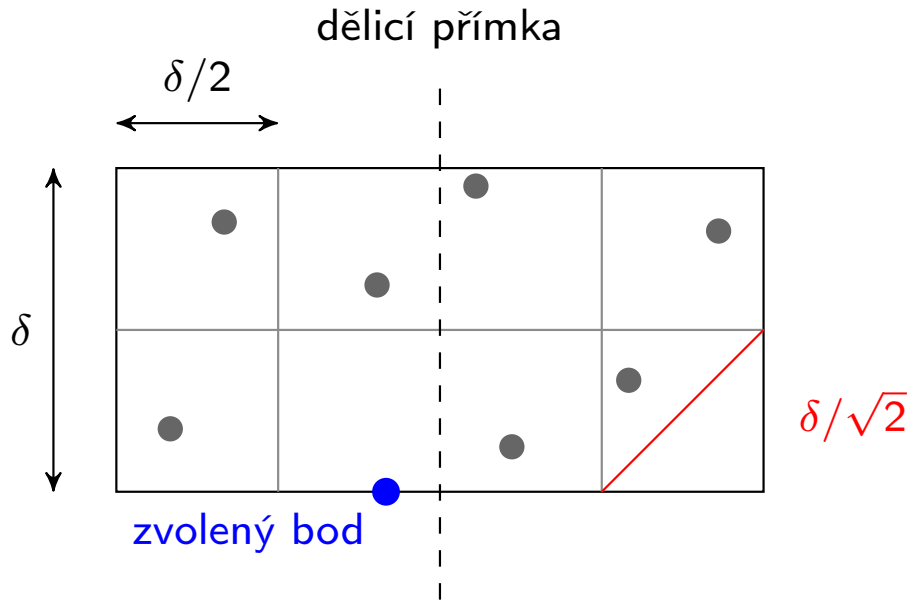
Na první pohled může být v pásu stále všech n bodů. Počet porovnání však omezuje geometrické pozorování:

Lemma 8 (Konstantní počet kandidátů)

Mají-li uvnitř levé i pravé poloviny všechny dvojice vzdálenost alespoň δ, stačí pro každý bod v pásu zkontrolovat nejvýše sedm následujících bodů v pořadí podle y.

Důkaz: obdélník $2\delta \times \delta$ nad zvoleným bodem rozdělíme na osm čtverců o straně $\delta/2$. Úhlopříčka čtverce měří $\delta/\sqrt{2} < \delta$, takže v jednom čtverci nemohou být dva body téže poloviny — to by odporovalo výsledku rekurze.

Osm čtverců



Do obdélníku se vejde nejvýše osm bodů včetně zvoleného, takže v pořadí podle y stačí zkontrolovat nejvýše sedm následovníků — další už mají y-ový rozdíl alespoň δ.

Algoritmus a zesílený návratový údaj

Nejbližší dvojice

1. Body rozdělíme podle x a rekurzivně vyřešíme obě poloviny.
2. Z menší nalezené vzdálenosti δ vytvoříme pás kolem hranice.
3. Body pásu procházíme podle y a každý porovnáme nejvýše se sedmi následujícími.
4. Vrátíme nejmenší ze vzdáleností z polovin a z pásu.

Kdybychom pás v každém volání znovu řadili podle y, stálo by skládání $O(n \log n)$ a celkem bychom dostali $O(n \log^2 n)$.

Stejně jako u počítání inverzí proto zesílíme návratový údaj: rekurzivní volání vrací zároveň své body seřazené podle y. Dvě seřazené posloupnosti sloučíme lineárně.

Výsledek

Věta 9

Nejbližší dvojici n bodů v rovině lze nalézt v čase $\Theta(n \log n)$.

Důkaz.

Každá dvojice leží buď celá v jedné polovině (řeší rekurze), nebo překračuje hranici; každá příčná dvojice bližší než δ leží v pásu a podle lemmatu se objeví mezi kontrolovanými kandidáty.

Po počátečním řazení za $O(n \log n)$ je práce jednoho volání mimo rekurzi lineární, tedy $T(n) = 2T(n/2) + \Theta(n)$.

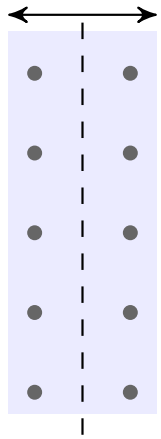
Geometrie místo hrubé síly

Samotné dělení nepomůže, pokud při skládání vyzkoušíme všechny dvojice přes hranici. Zrychlení přináší až *omezení hustoty*: v malé oblasti nemůže být mnoho bodů, aniž by dva byly blíž než dosavadní δ.

Shamos, Hoey (1975) — jedna ze zakládajících úloh výpočetní geometrie.

Kde se v důkazu skrývá past

Časté nedorozumění: „v pásu je nejvýše konstantně mnoho bodů“. To *není* pravda — v pásu jich může být všech n.



pás šířky 2δ

Správné tvrzení je slabší, ale úplně stačí:

každý jednotlivý bod má v pásu nejvýše sedm relevantních následovníků v pořadí podle y.

Celková práce v pásu je tedy $O(k)$ pro k bodů pásu, nikoli $O(k^2)$. Podobných argumentů o hustotě je v geometrii mnoho: omezený počet bodů v malé oblasti plyne z toho, že blízké dvojice už by odporovaly výsledku rekurze.

FFT jako další krok (rozšiřující studie)

Karatsuba snížil počet menších násobení. Pro velmi dlouhá čísla a polynomy lze jít dál: změnit reprezentaci tak, aby se složitá operace proměnila v jednoduchou.

Polynom stupně $< n$ lze zapsat

- *koeficienty* $(a_0, \ldots, a_{n-1})$ — násobení je konvoluce, $c_k = \sum_i a_i b_{k-i}$, tedy $\Theta(n^2)$;
- *hodnotami* v n různých bodech — násobení je po složkách, tedy lineární.

Pozor: součin polynomů stupně $< n$ má stupeň až $2n - 2$, takže jej určuje až $2n - 1$ hodnot. Oba činitele proto musíme vyhodnotit v alespoň $2n - 1$ společných bodech, teprve pak násobit a interpolovat zpět.

Násobení polynomů: malý příklad

Chceme $(1 + 2x) \cdot (3 + 4x)$ přes hodnoty. Součin má stupeň 2, potřebuje tedy tři body — volme $x = 0, 1, -1$.

x	$A(x) = 1 + 2x$	$B(x) = 3 + 4x$	$C(x) = A(x)B(x)$
0	1	3	3
1	3	7	21
-1	-1	-1	1

Interpolace zpět: hledáme $C(x) = c_0 + c_1 x + c_2 x^2$ splňující

$$c_0 = 3, \qquad c_0 + c_1 + c_2 = 21, \qquad c_0 - c_1 + c_2 = 1,$$

odkud $c_1 = 10$, $c_2 = 8$:

$$C(x) = 3 + 10x + 8x^2.$$

Násobení bylo triviální (*po složkách*); veškerá práce leží v převodu tam a zpět. FFT tyto dva převody zlevní na $\Theta(n \log n)$ — volbou bodů jako komplexních odmocnin z jedné.

Co dělá FFT

Rychlá Fourierova transformace převádí mezi reprezentacemi v čase $\Theta(n \log n)$. Rekurzivní varianta dělí koeficienty na sudé a liché:

$$A(x) = A_{\text{sud}}(x^2) + x\, A_{\text{lich}}(x^2),$$

tedy opět $T(n) = 2T(n/2) + \Theta(n)$.

Problém lze tedy někdy zrychlit přechodem do jiné reprezentace, v níž je požadovaná operace levná.



J. W. Cooley, J. W. Tukey.
An Algorithm for the Machine Calculation of Complex Fourier Series.
Mathematics of Computation 19(90) (1965), 297–301.

Společné návrhové motivy

1. **Počítej jen nezbytné podproblémy.**
   Karatsuba získá součet obou smíšených součinů jediným násobením; výběr pokračuje jen na jedné straně pivota.
2. **Zaplať za strukturu, která zmenší budoucí práci.**
   Medián mediánů stojí lineární čas, ale zaručí odstranění konstantní části vstupu.
3. **Zesil návratový údaj.**
   Nejbližší dvojice vrací také pořadí podle y.
4. **Použij vlastnost domény.**
   Omezení hustoty bodů nahradí kvadratické porovnávání konstantním počtem kandidátů.
5. **Změň reprezentaci.**
   FFT: skládání může být jednoduché v jiném pohledu na tatáž data.