

Delenie vecí napoly

Mišo Forišek

<misof@mfnotes.ksp.sk>
Department of Computer Science
Faculty of Mathematics, Physics, and Informatics
Comenius University, Bratislava, Slovakia

apríl 2008

1 Úvod

Návrh efektívnych algoritmov je z veľkej časti „skladačka“. Existuje $N < 30$ rôznych postupov, dátových štruktúr a trikov, ktoré dokopy pokryjú 99% všetkých efektívnych algoritmov.

Dnešná prednáška sa bude zaoberať jedným z týchto princípov. Ten náš nesie honosné meno „rozdeľ to tak od oka napoly“.

2 Binary search

2.1 Sprostá hra

Myslím si číslo od 0 do $2^k - 1$, uhádnite ho na čo najmenej áno/nie otázok.

Otázok treba k . Vo všeobecnom prípade, ak má vstup n možností, otázok treba $\lceil \log_2 n \rceil$. (Na k otázok vieme rozlíšiť 2^k možností, hľadáme najmenšie k také, že $2^k \geq n$.)

Odbočka: Keď triedime, máme $n!$ možností=permutácií, z ktorých si vyberáme. Preto treba $\log n! \sim n \log n$ otázok.

2.2 Pole

Takmer to isté je binárne vyhľadávanie v utriedenom poli. Tým, že pole veľkosti n delíme napoly, dospejeme k výsledku v čase $\log_2 n$.

2.3 Dynamická verzia

To, čo je binárne vyhľadávanie v statickom poli, vieme robiť aj na dynamických dátach – stromčekoch. Problém: potreba vyváženosti. Príklad vyváženého a nevyváženého stromu.

Zdôvodnené, že stačí vrcholy deliť napr. $1/3 : 2/3$, lebo po kažých dvoch krokoch zmenšíme rozsah na menej ako polovicu.

3 Rozdeľ a panuj

3.1 Mergesort

Rozdelíme pole na dve polovice, každú utriedime, spojíme dokopy. Strom rekurencie, ukázané, že je zložitosť $O(n \log n)$.

3.2 Geometrické konštrukcie

Konvexný obal: Rozdelíme na „ľavú“ a „pravú“ polovicu, rekurzívne nájdeme obaly, nájdeme spoločné dotyčnice.

Prienik polrovín: Nájdeme prienik jednej polovice, druhej polovice, a následne v $O(N)$ prienik dvoch konvexných útvarov.

Dva najbližšie body: Rozdelíme na „ľavú“ a „pravú“ polovicu, rekurzívne nájdeme dva najbližšie body vľavo a dva vpravo, minimum z ich vzdialeností označíme d , a v lineárnom čase prejdeme zhora nadol zvislý pás s polomerom d okolo miesta kde sa delilo na polovice.

4 Meet in the middle

4.1 Knapsack

Jeden z notoricky známych problémov. Základná verzia: Lupič má batoh s nosnosťou M . V miestnosti, kam sa vlámal, je n predmetov. Každý z nich má nejakú hmotnosť w_i a cenu c_i . Ktoré má pobrať, aby mali maximálny súčet cien?

Rozhodovací problém „dá sa dosiahnuť cena aspoň C ?“ je NP-úplný, lebo už SUBSETSUM je.

Možné riešenia:

- greedy nemusí byť optimálne
- aproximačné schémy
- ak sú váhy malé celé čísla, DP
- vyskúšať všetkých 2^n podmnožín

Len posledné je univerzálne funkčné. Problém: funguje tak po $n = 30$.

Finta ako ho zlepšiť po $n = 60$: Rozdelíme prvky na dve kopy po $n/2$ prvkov, pre každú zgenerujeme všetky možné súčty, utriedime a prelezieme.

Dôležité pozorovanie: Toto ide spraviť len raz!

(Nedá sa „urobiť to znova“ a dostať riešenie pre $n = 120$.)

4.2 Double DES

Úloha z IOI 2001 vo Fínsku.

Setting: Máme človeka, ten človek občas niečo šifruje, stále tým istým kľúčom. Sú známe algoritmy, ktoré používa na šifrovanie aj dešifrovanie, ale nepoznáme kľúč. Ku jednému šifrovanému textu C sa nám ale podarilo zistiť zodpovedajúci otvorený text P . Chceme zistiť kľúč, ktorým šifruje (aby sme si mohli čítať ďalšie jeho správy). Toto sa volá Known plaintext attack (KPA).

Náš chlapík, volajme ho Kleofáš, smie používať len 20-bitové šifrovanie, lebo viac je v jeho krajine nelegálne. Bojí sa ale, že by niekto pri KPA mohol vyskúšať všetkých 2^{20} kľúčov a zistiť, ktorý je ten správny. Preto šifruje každú správu $2\times$ za sebou, dvoma rôznymi kľúčmi. Takto teda legálne získal šifru s 40-bitovým kľúčom. . . aspoň si to myslí.

KPA proti double DESu sa dá spraviť technikou meet in the middle – všetkými kľúčmi K_1 zašifrujeme P , všetkými kľúčmi K_2 dešifrujeme C , nájdeme zhodu.

Existuje však technika triple DES (encrypt K_1 , decrypt K_2 , encrypt K_3), ktorá môže byť celkom fajn. Ale závisí od použitej šifry. Ak napríklad len sprost xorujeme, tak aj 17 kľúčov je to isté ako jeden.

4.3 Redukcia priestoru stavov

Keď hľadáme vo veľkom generovanom grafe najkratšiu cestu, môže sa oplatiť hľadať ju naraz z oboch koncov. Asymptoticky to zložitosť zlepšiť nemusí, ale pomôže.

Obrázkovo znázorniť priestory stavov preskúmané oboma prístupmi.

5 Binárne vyhľadávanie výsledku

5.1 K Best

Úloha: Mladá dáma vo finančných ťažkostiach sa rozhodla neriešiť ich vydajom, ale predajom rodinných šperkov. Má N šperkov, každý má svoju cenu a váhu. Chce si ich K nechať. Pritom chce, aby tie, čo jej ostanú, mali čo najväčšiu mernú cenu. Teda chce maximalizovať (súčet cien) deleno (súčet hmotností).

Zjavné greedy: utriedime podľa cena/hmotnosť.

Nefunguje: 7/1, 99/100, 9/10; $K = 2$. Najlepšie je zobrať prvý a tretí.

Riešenie: bsearch na hodnotu riešenia.

Pointa: Ak úlohu nevieme riešiť, dobrá otázka, ktorá nám môže pomôcť, je opýtať sa, či vieme overiť, či je niečo riešením / či existuje aspoň nejaké dobré riešenie.

5.2 Kino

Máme v kine N ľudí, každý chce niekam sadnúť. Hýbu sa naraz a neprekážajú si. V akom najkratšom čase to ide?

Presnejšie, štvorcová sieť s prekážkami, N začiatkov, N koncov.

Riešenie: Binárne vyhľadávame čas, pre konkrétny čas dostávame maximálne párovanie.

5.3 Ďalšie úlohy

Buoyancy (kvádre rôznych hustôt vo vode, chceme výšku hladiny), Drying (máme mokré veci, schnú o 1 unit za minútu, máme jednu pec, kde môžeme každú minútu usušiť jednej veci K unitov, nájdite najlepší čas).

Disclaimer. Tieto poznámky môžete voľne používať na ľubovoľné nekomerčné účely. Na akékoľvek komerčné využitie je potrebný súhlas autora. Ak v mojich poznámkach objavíte nejakú chybu, prípadne ich nejakým spôsobom viete doplniť, budem rád, ak mi dáte vedieť.

Pre potreby prípadného citovania má tento kus poznámok evidenčné číslo MF-0012.