

Importanța algoritmilor pentru problemele de la interviuri

Traian Rebedea

traian.rebedea@cs.pub.ro

Cuprins

- Introducere
- Despre interviuri
- Sesiunea practică
 - Probleme pentru interviuri
 - Niveluri de dificultate
 - Rezolvări
- Sfaturi
- Concluzii

Despre mine

- Doctorand în prelucrarea limbajului natural (în special **conversații** online), cu experiență bună în ML, IR și IA în general
- Organizat laboratoare Analiza Algoritmilor și Proiectarea Algoritmilor (de la zero)
 - Cu ajutorul mai multor colegi de-a lungul timpului 😊
- Predau cursul de Proiectarea Algoritmilor la CA & FILS engleză

Utilitatea algoritmilor

- Deși algoritmi sunt folosiți mai mult de către dezvoltatorii software, cunoașterea lor este utilă în majoritatea domeniilor IT&C
 - Într-o măsură mai mare sau mai mică
- Sunt folosiți și în rețele, telecomunicații, proiectare hardware, robotică, automatizări, ...
- Algoritmi ajung să fie stăpâniți bine doar prin ***experiență***
 - Algoritmi clasici, probleme clasice
 - Rezolvarea cât mai multor probleme
 - Studiul algoritmilor avansați / specializați

Motivația temei

- Mulți studenți sunt interesați de întrebările primite la interviuri
- Majoritatea ***companiilor de top*** nu caută “programatori Java”, ci ***dezvoltatori talentați!***
- Cunoașterea algoritmilor...

Ce fel de interviuri?

- Dezvoltare software
- Acele companii care nu caută angajați pentru o anumită tehnologie, ci pentru ***anumite calități***
 - Gândire logică, algoritmi, rezolvare de probleme, testare, validare, concepte fundamentale înțelese bine
 - Inclusiv posibilitatea de adaptare la orice tehnologie
- Atât din România, cât și din afară

Cum se desfășoară?

- La interviurile telefonice: rezolvarea se face într-un tool online pentru editare colaborativă (collabedit, gdocs, etc.)
- La interviurile F2F (on-site): rezolvarea se face pe whiteboard, hârtie, etc.
- Deci ***nu aveți la dispoziție*** compilator, IDE (auto-complete), documentație API, net, etc.

Cum se desfășoară?

- Depinde de companie, candidat, experiență anterioară, poziție, etc.
 - Presupunem că vorbim despre SDE / SDET
 - Însă există mai multe tipuri de SDE
- Ne vom concentra pe SDE, “generalist”, fără prea multă experiență ca angajat, firmă ce caută oameni deștepți
 - Internship-urile sunt foarte utile (chiar necesare)!
- Independent de tehnologii și limbaje de programare
- Pentru specialiști, interviurile sunt un pic diferite

Interviu telefonic

- 1-2(-3) runde de aproximativ 45 de minute
- 1-2 probleme per rundă
- Dificultate ușoară-medie
- De obicei, pentru internship la companiile din afara RO

- Pot fi și întrebări “comportamentale”
- Aveți timp să puneți întrebări la final
- Nu trebuie să întrebați cum v-ați descurcat la final
 - Însă puteți întreba pe parcurs dacă există (sau vor) o soluție **mai bună** decât cea propusă de voi

Interviu on-site

- 4-5 runde de 45 de minute
- Sunt probleme mai dificile și mai variate
- De obicei, pentru job full-time in afara RO

- 2-3 runde sunt mai orientate către rezolvarea de probleme, algoritmi, etc.
- Întrebări “comportamentale”, de proiectare, sisteme de operare, etc.

Abordare generală

- Puneți întrebări ca să fiți siguri că ați înțeles problema ce trebuie să fie rezolvată
- Căutați rezolvarea de complexitate **optimă**
- Dacă nu o găsiți într-un timp rezonabil, **nu intrați în panică** și propuneți o altă rezolvare mai slabă
 - De multe ori, se acceptă soluții mai puțin eficiente
- Fiți clari în explicații și rezolvare, **vorbiți pe măsură ce implementați soluția**, negociați diverse abordări, spuneți dacă aveți mai multe variante
- **Testați**-vă soluția cu 1-2 exemple, identificați cazuri extreme, gândiți-vă la dezavantaje, etc.
- Reparați greșelile făcute

Cum abordezi o problemă?

- Dacă o știi sau este foarte similară cu o altă problemă cunoscută
 - Atunci e simplu
- Altfel
 - Determinați probleme similare cu cea de rezolvat
 - Rezolvați probleme din ce în ce mai mari pe baza celor de dimensiune mai mică (greedy, PD, bkt)
 - Simplificați, rezolvați și apoi generalizați din nou
 - Gândiți-vă dacă anumite structuri de date vă simplifică rezolvarea (hash, heap, arbori binari, liste, vectori,...)
 - Încercați câteva exemple și generalizați o soluție

Tipuri de probleme

- O varietate destul de mare
- Structuri de date elementare (vectori, stringuri, liste, etc.)
- Arbori și grafuri
- Operații pe biți
- “Brain teasers”
- POO
- Recursivitate
- Sortări și căutări
- Probleme matematice
- Probleme NP-complete (bkt, căutări, etc.)
- ...

Sesiunea practică

- Am încercat să adun o serie de probleme reprezentative
- Am încercat să fie probleme pe niveluri de complexitate diferite
 - Teoretic, dificultatea crește cu id-ul problemei
- La un interviu, sunt șanse să primiți 1-2 astfel de probleme / sesiune
- Voi veți lucra în echipe să rezolvați problema
 - Alternative de rezolvare, ideea aleasă, pseudocod, complexitate (aprox. 10 minute / problemă)
 - Eventual și cod într-un limbaj la alegere, poate rămâne temă de casă

Problema 1

- Implementați o căutare a unui element într-un vector care a fost sortat, dar și rotit la dreapta cu un număr oarecare de poziții.
- Ex: **[8, 9, 10, 3, 4, 5, 6, 7]**

Hint-uri

- Complexitate?
- Probleme similare?

Întrebări suplimentare

- Ce se întâmplă dacă aveți elemente duplicate?
- Mai puteți atinge aceeași complexitate?

Sursa 1

- <http://xorswap.com/questions/77-implement-binary-search-for-a-sorted-integer-array-that-has-been-rotated>

Problema 2

- Având un vector cu numere pozitive, determinați cea mai mare sumă ce poate fi obținută prin adunarea numerelor aflate în poziții non-consecutive.
- Dacă se alege numărul $v[i]$, nu aveți voie să folosiți $v[i-1]$ sau $v[i+1]$

Hint-uri

- Încercați pe exemple mici
- Complexitate intuită?
- Tehnici clasice?

Sursa 2

- <http://www.glassdoor.com/Interview/Facebook-Interview-RVW1491822.htm>
- http://www.glassdoor.com/Interview/Find-the-maximum-sum-possible-in-an-array-of-positive-integers-by-selecting-the-elements-in-such-a-way-that-no-two-elements-QTN_250824.htm

Problema 3

- Cum ați salva și căuta (cu auto-complete, fără corecții automate) peste 1 milion de nume?

Hint-uri

- Structuri de date

Sursa 3

- http://www.glassdoor.com/Interview/How-would-you-store-and-search-1-million-names-QTN_256940.htm

Problema 4

- Determinați 3 elemente dintr-un vector de numere întregi a căror suma este 0.

Hint-uri

- Soluția banală este evidentă
- Știți o problemă similară, eventual mai simplă?
- O soluție mai bună ar trebui să fie detectată ușor
- Soluția cea mai bună este în $O(n^2)$

Sursa 4

- http://www.glassdoor.com/Interview/Given-an-array-of-integers-check-whether-there-are-3-numbers-that-sum-up-to-0-QTN_258361.htm

Problema 5

- Determinați maximul dintre două numere întregi fără a folosi instrucțiunile *if-else* și nici operatori de comparație.

Hint-uri

- Reformulați soluția ca să respecte cerințele

Soluție

Evaluate: $a - k * (a - b)$

```
int getMax(int a, int b) {  
    int c = a - b;  
    int k = (c >> 31) & 0x1;  
    int max = a - k * c;  
    return max;  
}
```

Sursa 5

- Cracking the Coding Interview

Problema 6

- Considerând că avem următoarea codificare a caracterelor în numere:

a -> 1, b -> 2, ... , z -> 26

- Fiind dat un număr întreg, afișați numărul total de traduceri posibile în string-uri conform codificării date.

– Ex: 11223

Hint-uri

- Putem considera subprobleme?

Întrebări suplimentare

- Pot exista și numere pentru care nu există traduceri posibile?
- Cum facem dacă vrem să și afișăm toate traducerile posibile?

Sursa 6

- http://www.glassdoor.com/Interview/Suppose-we-can-translate-numbers-into-characters-1-and-gt-a-2-and-gt-b-26-and-gt-z-given-an-integer-for-example-11223-output-QTN_291436.htm

Problema 7

- Având 9 bile, dintre care una cântărește mai puțin decât celelalte, și o balanță, cum puteți determina care dintre bile este cea mai ușoară?

Întrebări suplimentare

- Care este complexitatea?
- Generalizați pentru numere mai mari

Sursa 7

- http://www.glassdoor.com/Interview/You-have-9-balls-one-of-which-weighs-less-than-the-others-but-identical-in-appearance-If-you-had-one-weighing-scale-how-QTN_248854.htm

Problema 8

- Determinați numărul de cifre '2' conținute de toate numerele între 0 și n .

Hint-uri

- Ce complexitate vrem? Mai mică decât?

Sursa 8

- Cracking the Coding Interview

Problema 9

- Avem un generator de numere aleatoare care trimite fiecare valoare nou generată unei funcții ce trebuie să mențină mediana tuturor numerelor primite până la orice pas.

Hint-uri

- Care sunt soluțiile banale?
- Folosiți deștept procesările anterioare
- Structuri de date mai bune?

Sursa 9

- Cracking the Coding Interview

Problema 10

- Determinați cele mai mari 1 milion de numere dintr-un vector cu un miliard de numere.

Întrebări suplimentare

- Puteți păstra tot vectorul în memorie?

Hint-uri

- Soluția banală?
- Se poate mai bine!
- Structuri de date
- Probleme similare, algoritmi mai buni?
 - Selection rank?

Sursa 10

- Cracking the Coding Interview

Sfaturi

- Este esențial să vă **antrenați**
 - TopCoder (Arena), Infoarena
 - Dar și cu pixul pe hârtie, cu limite de timp și probleme similare
- Sunt importante și întrebările comportamentale
- **Nu puteți memora** algoritmi sau soluții
 - Trebuie să le înțelegeți corectitudinea
- **Nu vă grăbiți**, nu vorbiți foarte mult/puțin
- Experiența la interviuri nu poate fi decât utilă, chiar dacă primul răspuns nu este pozitiv

Considerații finale

- Contează și alte amănunte
 - **Recomandări** ale altor angajați
 - Răspunsurile la întrebările “comportamentale”
- Intrebări de CS in general: structuri de date, sisteme de operare, (mai rar logică, statistică și sunt bonus)
- **Motivația** voastră, **proiectele personale**, ...
- Este foarte important să și **codați pe tablă** soluțiile corect și eficient într-un limbaj la alegere

Concluzii

- Întrebările de la interviuri conțin multe elemente de algoritmi
 - În special pentru un SDE(T) internship la companiile mari
- Întrebările ***nu sunt foarte dificile***
- Necesită o ***înțelegerea foarte bună a algoritmilor de bază***, dar și experiență
- ***Exersați*** pe hârtie sau pe Topcoder Arena direct, pentru a nu folosi un IDE, compilator, etc.
- Nu subestimați ***emoțiile***
- Nu subestimați întrebările comportamentale: evitați aroganța, individualismul, etc.

În final

- Avem din ce în ce mai multe firme mari în țară, care lucrează la proiecte interesante
- Sunt din ce în ce mai mulți români (inclusiv absolvenți de A&C) care lucrează în străinătate
 - Discutați cu ei pentru sfaturi și recomandări
- Nu trebuie să fi fost olimpic, deși acest lucru este evident un atu
- Vă pot îndruma înspre ei sau spre recruteri

Link-uri

- Carte: Cracking the Coding Interview
 - <http://www.careercup.com/book>
- Site-uri
 - <http://www.careercup.com/>
 - <http://infoarena.ro/missing-numbers>
- Blog-uri
 - <http://infoarena.ro/blog/sfaturi-pentru-interviuri>
 - <http://stiinte.ub.ro/admitere/53-c-anunturi-generale/282-google-zurich-1>
 - <http://www.kmkz.ro/de-pe-teren/reportaje/cum-am-ajuns-la-interviu-la-facebook/>