

Algoritmer, TAIK12, Individuell inlämningsuppgift

Cecilia Sönströd, Niclas Ståhl, Vladimir Tarasov

5 maj 2022

1 Inledning

Denna inlämningsuppgift ingår i kursen Algoritmer, TAIK12, på Civilingenjörsutbildningen i Datateknik, år 1. Syftet med inlämningsuppgiften är att ge en fördjupning inom området algoritmer. Genom att studenten själv skall implementera och på ett kontrollerat sätt utvärdera en algoritm introduceras ett vetenskapligt arbetssätt. Uppgiften tränar dessutom skriftlig framställning genom att resultatet kommuniceras i form av en laborationsrapport.

2 Uppgiften

Varje student skall välja en algoritm från listan nedan. Val av algoritm meddelas via mail till Cecilia. Uppgiften består av att först läsa in sig på algoritmen och sedan implementera samt testköra den på ett kontrollerat sätt i syfte att studera implementationens komplexitet. Arbetet redovisas genom en skriftlig rapport (cirka 4-6 sidor).

3 Detaljer

Vid implementering är det mycket viktigt att lämpliga datastrukturer väljs. Många algoritmers komplexitet är helt beroende av vald representation. Avseende kodkvalitet, så måste implementationen av algoritmens centrala idéer dels följa den pseudokod man utgått ifrån och dels vara tydligt och korrekt genomförd, så att resonemang om komplexitet etc. går att utföra. Implementering sker i språket C och ni skall kunna lämna in en körbar fil där man kan variera input till algoritmen. All kod i denna uppgift skall skrivas egenhändigt; det är alltså inte tillåtet att använda sig av kodsntuttar från externa källor.

Huvudsyftet med laborationen är att undersöka algoritmernas komplexitet. Detta ställer naturligtvis krav på att man har möjlighet att variera input. I vissa fall, när det också finns en brute-force algoritm som löser samma problem, skall även denna implementeras och jämförelser göras. För vilka algoritmer detta gäller, se nästa sida.

Rapporten kan utformas på något olika sätt, men bör följa nedanstående mall där varje punkt motsvarar ett avsnitt:

1. **Inledning.** En kort beskrivning av studiens övergripande syfte, inklusive kort presentation av det algoritmiska problemet som studeras.
2. **Algoritmen eller algoritmerna i detalj.** Kopiera inte bara från källorna utan se till att er beskrivning här blir så bra och detaljerad som möjligt. Ni kan använda både beskrivningar i pseudokod och exempel på hur algoritmen arbetar. Se dock till att beskrivningen innehåller en generell förklaring, dvs. inte baserad endast på en exempelexekvering.
3. **Designval.** Vilka konkreta val gjordes vid implementeringen; t.ex. avseende datastrukturer och de delar av algoritmer som inte specificerats detaljerat i pseudokoden? Motivera val och koppla till teori, t.ex. avseende hur val av representation påverkar tids- och/eller minneskomplexitet.
4. **Experimentplanering.** Vilken input användes? Vilka körningar genomfördes? Motivera samtliga val utförligt, exempelvis genom att förklara hur input designats för att testa *best*, *average* resp. *worst case* till de olika algoritmerna.

5. **Resultat.** Uppmätta tider eller andra kvantitativa resultat, exempel antal exekveringar av den kritiska operationen. Fundera noga igenom vilka inputstorlekar det är vettigt att jämföra och hur resultaten bör presenteras. I de fall ni har icke-deterministiska element i era experiment, t.ex. slumpmässig input, kör upprepade exekveringar med samma inputstorlek och rapportera medelvärden, samt eventuellt standardavvikelser.
6. **Analys.** Resultaten kopplade till och belysta utifrån teorin. Här bör man resonera om de resultat som körningarna gav i relation till algoritmens teoretiska komplexitet, samt reflektera lite över sitt experimentupplägg.
7. **Slutsatser och diskussion.**

Appendix: Programkod och körningsexempel.

4 Examination

Den skriftliga rapporten och implementeringen bedöms tillsammans med betyget U eller G. Som vanligt finns 3 examinationstillfällen, där det första tillfället är 2022-05-27.

5 Genomförande

Uppgiften genomförs individuellt.Handledning sker vid två tillfällen, med bokning i Canvas:

Handledning 1: 2022-05-12. Fokus på val av datastruktur och experimentplanering. Handledning 2: 2022-05-19. Fokus på rapporten, struktur och innehåll.

Observera alltså att handledning inte ges på kodnivå. Det är ert eget ansvar att få till en fungerande implementation, vilket innebär att man bör välja en uppgift som man känner att man reder ut kodmässigt. All kod skall skrivas egenhändigt, vilket innebär att det är strängt förbjudet att hämta kodsuttag från nätet eller att dela kod med varandra.

6 Algoritmer

Alla sidhänvisningar är till [\[Lev11\]](#).

1. Binärsökning och interpolationssökning (s. 177 – 178, 187 – 189)
2. Quickselect med Lomutopartition (s. 184 – 186) *
3. Närmsta paret med divide-and-conquer (s. 218 – 221) *
4. Konvexa höljet med divide-and-conquer (s. 221 – 223) *
5. Quicksort med valfri förbättring (202 – 207):
 - olika strategier för val av pivotelement
 - implementering av insertion sort i botten av rekursionen
 - modifierad partitionering (exempelvis i tre delar)
6. Horspools algoritm för strängmatchning (s. 285 – 288)*
7. Boyer-Moores algoritm för strängmatchning (s. 289 – 292)*
8. Öppen och sluten hashing (s. 295 – 300)
9. Bucket sort med hashning (se pseudokod på länk)
10. B-träd med insättning och uppslag (s. 302 – 305)
11. Optimala binära sökträd (s. 323 – 328)
12. Warshalls algoritm (s. 330 – 334)
13. Floyds algoritm (s. 334 – 337)
14. Prims algoritm (s. 344 – 348)
15. Kruskals algoritm (s. 351 – 357)

16. Dijkstras algoritm (s. 359 – 363)
17. Huffmans algoritm. (s. 364 – 368)
18. Backtracking. Valfri tillämpning som inte finns i boken. (s. 450)
19. Branch-and-bound. Gärna ett exempel som inte finns i boken. (s.458)
20. TSP. Avancerade lösningsstrategier/algoritmer som inte finns i boken.*
21. SAT. Någon avancerad lösningsstrategi som inte finns i boken.*

För algoritmer markerade med * ingår implementation av brute-force algoritm med samma representation.

Referenser

- [Lev11] Anany Levitin. *Introduction to the Design and Analysis of Algorithms*. Pearson Education, third edition, 2011.