

Derde huiswerkopgave Complexiteit 2025

Inleveren: maandag 28 april 2025

Maak de uitwerking in **L^AT_EX**!

Inleveren via Brightspace

Geef een *duidelijke toelichting/uitleg* bij al je antwoorden!

In deze huiswerkopgave ga je experimenteren met verschillende series stapgroottes in Shell-sort. Op de website vind je een implementatie van Shellsort (in C++), en een makefile voor het compileren. Door vervolgens bijvoorbeeld `./shellsort 10000 10` aan te roepen¹ wordt 10 keer een willekeurig rijtje van 10.000 (verschillende) getallen gegenereerd (dus van *lengte* 10.000) en gesorteerd met alle gedefinieerde series stapgroottes. Tijdens het sorteren wordt het aantal (array)vergelijkingen bijgehouden. Het programma geeft standaard als uitvoer, per serie stapgroottes, het aantal vergelijkingen dat het minimaal, maximaal en gemiddeld heeft moeten doen.

De tijdens het college besproken series stapgroottes van Shell [1] en Hibbard [2] zijn al inbegrepen, net als de serie met alleen stapgrootte 1 (m.a.w. Insertion sort).

Voer de volgende experimenten uit en schrijf hier een (kort) verslag over.

a. (20 punten)

De oorspronkelijke keuze van Donald Shell was om telkens te delen door een factor 2. Pas het programma aan om ook andere factoren dan 2 te gebruiken en onderzoek welke factor de beste resultaten geeft:

- Gebruik factoren 2, 3, en minstens zes ertussen. Ze hoeven niet evenredig te zijn verdeeld.
- Genereer in dit experiment rijtjes van drie verschillende lengtes: een tweemacht, een driemacht, en een getal dat geen veelvoud is van 2 of 3 (vermenigvuldig bijvoorbeeld grotere priemgetallen totdat je getal groot genoeg is). Zorg ervoor dat elke lengte minstens 100.000 is.
- Geef per lengte één tabel, met daarin per factor (in ieder geval) het gemiddelde, minimale en maximale aantal vergelijkingen. Geef ook duidelijk aan (bijvoorbeeld in de caption) welke lengte je hebt gebruikt en hoeveel rijtjes je hebt gegenereerd (d.w.z., welke parameters je hebt meegegeven aan het programma). Genereer voor elke lengte minstens 10 rijtjes.
- Geef ook op een overzichtelijke manier per lengte en per factor de resulterende stapgroottes.
- Bespreek de resultaten. Geef in ieder geval per lengte aan welke factoren het het beste deden, welke juist niet, en waarom je denkt dat dit het geval is.

¹In een terminal, maar je kan de code uiteraard aanpassen voor andere situaties.

b. (20 punten)

Tijdens het college zijn de series stapgroottes van Shell en Hibbard uitvoerig besproken, maar zijn ook een paar andere genoemd. Op internet is ook van alles te vinden, bijvoorbeeld op DBLP² of op Wikipedia³. Pas het programma aan om ook andere series te genereren en experimenteer hiermee:

- Implementeer drie al bekende series naar keuze, en één serie dat je zelf hebt bedacht. Beschrijf deze series en geef voor de bekende series aan waar ze vandaan komen, graag met een referentie naar het oorspronkelijke artikel (zoals [1, 2]). Je kan de artikelen bijvoorbeeld zoeken op DBLP of op Google Scholar⁴. Geef bij de zelfbedachte serie aan hoe deze is geconstrueerd en (zo mogelijk) hoe je erop bent gekomen.
- Vergelijk de prestaties van de vier nieuwe series stapgroottes met de drie die al zijn gegeven (Shell, Hibbard, Insertion sort). Geef je resultaten weer in een tabel zoals bij **a**. Gebruik weer minstens 10 rijtjes van lengte minstens 100.000 en geef deze waardes duidelijk aan. Deze keer hoeft je experimenten maar uit te voeren op één lengte. Geef per serie stapgroottes (in ieder geval) het gemiddelde, minimale en maximale aantal vergelijkingen, en noteer de concrete stapgroottes voor de bekeken lengte. Bespreek vervolgens de resultaten: geef in ieder geval aan hoe de series het in jouw experiment deden en waarom je denkt dat dit het geval is.

Insertion sort doet er heel lang over voor lange rijtjes. Het aantal vergelijkingen kan je afschatten op $\frac{1}{4}n^2$; zie college 5. Geef dit wel even aan in de caption.

- We loven een reep chocola uit voor de best presterende en onderbouwde zelfbedachte serie ;)⁵

c. (10 punten)

Voer de series stapgroottes uit opgave **b** ook uit op oplopend en aflopend gesorteerde rijtjes. Geef je resultaten weer in twee tabellen, zoals beschreven bij opgave **b**, en vergelijk de waardes met elkaar en met de resultaten van opgave **b**. Voor Insertion sort kan je wederom de resultaten uit college 5 gebruiken: $n - 1$ voor het oplopend gesorteerde rijtje en $\frac{1}{2}n(n - 1)$ voor het aflopend gesorteerde rijtje.

Opmerkingen bij het programmeren

- De geleverde code gebruikt de klasse `vector` in plaats van C-arrays: <https://cplusplus.com/reference/vector/vector/>.
- De stapgroottes moeten in aflopende volgorde staan bij de aanroep van de functie `shellsort`. Als je een rijtje stapgroottes in oplopende volgorde opbouwt, kan je `reverse` gebruiken om het om te draaien: <https://cplusplus.com/reference/algorithm/reverse/>, inclusief voorbeeld. Voor sommige rijtjes stapgroottes kan `sort` (<https://cplusplus.com/reference/algorithm/sort/>) ook nuttig zijn.
- Je kan de uitvoer van het programma, in de functie `run`, aanpassen om (delen van) L^AT_EX-tabellen te produceren.

²<https://dblp.uni-trier.de/search?q=shellsort>

³<https://en.wikipedia.org/wiki/Shellsort>

⁴<https://scholar.google.com>

⁵Prestaties in door ons uitgevoerde experimenten, onderbouwing naar ons inzicht.

Referenties

- [1] Shell, D.L., 1959. A high-speed sorting procedure. Communications of the ACM, 2(7), pp.30-32.
- [2] Hibbard, T.N., 1963. An empirical study of minimal storage sorting. Communications of the ACM, 6(5), pp.206-213.