Proiectarea algoritmilor
Tema de casă

Profesor: Costin Badică
Asistenţi: Alexandra Vultureanu-Albişi, Andrei Andreiana,
Ionuţ Murareţu, Ruxandra Luţan

Semestrul II, 2023-2024

Abstract

Acest document introduce obiectivele şi metodologia de realizare a temelor pentru cursul de Proiectarea algoritmilor. Documentul conţine, de asemenea, şi descrierea livrabilelor.

1 Introducere

Obiectivul temelor este de a dezvolta software pentru a experimenta astfel înțelegerea algoritmilor. Sarcinile sunt axate pe dezvoltarea abilitaților pentru o bună programare a algoritmilor de bază, acoperind proiectarea, codarea, documentarea și prezentarea rezultatelor. Realizarea temei implică să furnizati un set de livrabile, ce include:

- Raport tehnic, .doc/.docx sau LaTeX [1] (Overleaf tutorial).
- Cod sursă în limbajul C (obligatoriu) și limbajul Python (OPTIONAL pentru puncte bonus)
- Date experimentale non-triviale mari și foarte mari generate aleator.
- Codul sursă în limbajul C (obligatoriu) și/sau în limbajul Python al algoritmilor pentru generarea datelor de intrare.

2 Aspecte generale

2.1 Enunțul problemei

Un pescar explorează o regiune de coastă bogată în homari, fiecare având propria sa dimensiune (în centimetri) și valoare (în monede de aur). Plasa pescarului are o capacitate limitată, exprimată în numărul total de centimetri pe care îi poate conține. Având o listă detaliată cu

dimensiunile și valorile homarilor disponibili în acea regiune, sarcina este să se elaboreze o strategie prin care pescarul să selecteze homarii în așa fel încât să maximizeze valoarea totală a capturii, respectând în același timp limita de capacitate a plasei. Trebuie să se decidă care homari să fie incluși în plasă și care să fie lăsați, astfel încât suma valorilor homarilor selectați să fie cea mai mare posibilă, fără ca suma dimensiunilor acestora să depășească capacitatea plasei.

Imaginați-vă un scenariu în care un pescar are oportunitatea de a alege dintre o selecție de homari, fiecare cu o dimensiune și o valoare specificate, pentru a umple plasa sa care are o capacitate maximă. Scopul pescarului este de a maximiza valoarea totală a capturii sale fără a depăși limita de dimensiune a plasei.

Iată un exemplu:

Homari disponibili:

- **Homarul A: Dimensiune = 4 cm, Valoare = 20 monede de aur**
- **Homarul B: Dimensiune = 3 cm, Valoare = 15 monede de aur**
- **Homarul C: Dimensiune = 2 cm, Valoare = 10 monede de aur**
- **Homarul D: Dimensiune = 5 cm, Valoare = 25 monede de aur**

Capacitatea plasei: 10 cm

Provocarea este de a selecta combinația de homari care maximizează valoarea totală fără a depăși o dimensiune totală de 10 cm.

O posibilă soluție ar implica alegerea homarilor A și C, oferindu-ne o dimensiune totală de 6 cm (4 cm + 2 cm) și o valoare totală de 30 monede de aur (20 + 10). Totuși, o soluție mai bună ar fi să alegem homarii B, C și D, care împreună au o dimensiune totală de 10 cm (3 cm + 2 cm + 5 cm) și oferă o valoare totală mai mare de 50 monede de aur (15 + 10 + 25). Această combinație umple exact capacitatea plasei și maximizează valoarea capturii.

2.2 Modalitatea de notare

Notarea temelor se face ținând cont de toate elementele prezentate în acest document:

- Structura și continutul raportului tehnic trebuie să respecte cerințele specificate în secțiunea 4.
- Structura și conținutul codului sursă trebuie să respecte cerințele specificate în secțiunea 3.2.
- Conținutul datelor experimentale și rezultatele trebuie să respecte cerințele de scriere în secțiunea 3.3.

Vor fi adăugate puncte suplimentare pentru o implementare corectă a temei in Python.

3 Livrabile

Trebuie sa realizați trei tipuri de livrabile pentru tema de casă pe care o veți primi:

i) Raportul tehnic

ii) Codul sursă și documentația codului

iii) Date experimentale și rezultate

3.1 Raportul tehnic

Trebuie să descrie succint, concis și clar munca fiecăruia și rezultatele obținute. Descrierea trebuie să conțină enunțul problemei, cum a fost realizat programul și ce s-a obținut din secțiunea 4.

3.2 Codul sursă

Codul sursă trebuie să conțină sursele complete ale aplicației, incluzând fișierele sursă, fișierele antet, makefile-uri sau alte fișiere de configurare necesare. Codul trebuie să se compileze și să se construiască folosind open tools și nu trebuie să conțină dependențe de medii integrate de dezvoltare. Codul trebuie să fie trimis ca un repository public pe GitHub[1]. Vă rugăm să utilizați contul instituțional pentru autentificarea pe GitHub.

Trebuie îndeplinite următoarele cerințe minime pentru codul sursă. Acestea vor influența direct notarea temei dumneavoastră.

- Trebuie să furnizați instrucțiuni clare pentru compilarea codului și construirea fisierelor binare ale aplicației din codul sursă. Instrucțiunile ar trebui incluse în fișierul README.md al repository-ului.
- Codul trebuie să fie bine organizat în module cu interfețe clare.
- Codul trebuie să fie bine aranjat, cu indentare și o convenție clară de denumire a variabilelor, funcțiilor, claselor și pachetelor.
- Codul ar trebui să fie bine comentat.

3.3 Experimente și rezultate

În această secțiune trebuie explicată metoda utilizată pentru testarea algoritmilor, precum și rezultatele experimentale obținute. Ar trebui să vă verificați aplicația cu date de intrare non-triviale. Ar trebui să furnizați cel puțin 10 seturi de date de intrare non-triviale și ieșirile corespunzatoare produse de aplicația dvs. Codul sursă pentru generarea datelor de intrare non-triviale va fi si el furnizat.

Rezultatele experimentale vor conține:

[1] https://github.com/

- O descriere a datelor de ieșire obținute prin rularea algoritmilor, precum și metoda utilizată pentru a testa / verifica dacă datele de ieșire sunt corecte în raport cu specificațiile algoritmului.
- Opțional, timpul de execuție al algoritmilor, pentru fiecare mulțime de date de intrare.

De asemenea, trebuie să prezentați rezultatele (și opțional timpul de execuție înregistrat) ale experimentelor dvs. într-un mod semnificativ pentru cititor. Metoda de prezentare este lăsată la alegere.

4 Raportul tehnic

Raportul tehnic trebuie să fie tipărit utilizând LaTeX sau .doc/.docx și furnizat în formă electronică (PDF și surse în cazul LaTeX). Pentru documentație despre LaTeX puteți consulta referința [3]. Raportul tehnic trebuie împărțit în mai multe secțiuni, după cum urmează:

i) Coperta

ii) Enunțul problemei

iii) Datele experimentale

iv) Proiectarea experimentală a aplicației

v) Rezultate & concluziile

4.1 Coperta

Aceasta este o secțiune de o pagină care conține titlul temei de laborator, numele studentului / studenților, grupa, anul și secțiunea în care studentul este înscris.

4.2 Enunțul problemei

Această secțiune este introducerea raportului dvs. tehnic. Ar trebui să descrie clar sarcina temei de laborator.

4.3 Algoritmi

Această secțiune ar trebui să conțină descrierea pseudo-codului algoritmilor folosiți pentru tema primită. Pseudo-codul ar trebui să utilizeze formatul introdus în [2]. Un exemplu este prezentat în figura 1.

Mai mult, în această secțiune trebuie de asemenea inclusă o analiză a cerințelor de memorie și a complexității computaționale pentru fiecare algoritm creat pentru temă.

INSERTION-SORT(A)
1. **for** $j = 2$,$length[A]$ **do**
2. $key = A[j]$
3. $\triangleright$ Insert $A[j]$ into the sorted sequence $A[1 \dots j-1]$
4. $i = j - 1$
5. **while** $i > 0$ and $A[i] > key$ **do**
6. $A[i+1] = A[i]$
7. $i = i - 1$
8. $A[i+1] = key$

Figure 1: Example for typesetting an algorithm.

4.4 Date experimentale

Aplicația dvs. trebuie să abordeze, de asemenea, metoda de generare automată a datelor de test de intrare non-triviale. Datele non-triviale pot fi obținute prin generarea aleatorie de seturi de date mari și foarte mari, de exemplu, vectori de lungime 10^3-10^8.

4.5 Rezultate & Concluzii

Rezultatele trebuie să includă descrierea datelor de ieșire obținute folosind implementările C și Python ale fiecărui algorithm, precum și o analiză comparativă efectuată pe aceleași mulțimi de date. De exemplu, pot fi create tabele și/sau grafice în care să fie comparete media timpilor de execuție (timpul cpu) corespunzători mai multor rulări pe seturile de date de dimensiuni din ce în ce mai mari. Graficele pot fi create cu alte instrumente, de exemplu Microsoft Excel, Matlab, Python (matplotlib), etc. Rezultatele trebuie să fie în mod explicit descrise și explicate.

Această secțiune trebuie de asemenea să conțină propriile voastre concluzii după efectuarea sarcinilor. Câteva sugestii despre ce se poate prezenta aici: un rezumat al rezultatelor obținute, care au fost cele mai provocatoare și mai interesante părți însoțite de argumentări, direcții viitoare privitoare la extinderea studiului pe termen scurt sau lung etc.

5 Termen predare temă

Termenul **limită strict** pentru predarea temei este **marți, 30 mai 2024**, la sfârșitul zilei. Trebuie să livrați raportul ca un document **PDF** în care **TREBUIE SĂ INCLUDEȚI link-ul repository-ului**.

References

[1] Leslie Lamport, *LaTeX: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.

[2] Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*. MIT Press, 3rd Edition, 2009.

[3] LaTeXproject site, http://latex-project.org/, accessed in April 2013.