Structuri de date

Lect. Dr. Marius Dumitran

Organizatorice

- Notare
- Laboratoare / laboranți
- Curs live

Notare

- 40% laborator
  - **Nota minim 5!!**
- 20% seminar
  - Prezență, activitate și teme

- 40% examen
  - Examen scris (10-11 iunie ?)
  - **Nota minim 5!!**

- 10% Kahoot
  - Vom face teste uneori (probabil)... vedem cum e cu hibridul (azi vom avea exemplu)
- Media minim 4:50

Notare

- 40% laborator
  - **Nota minim 5!!**
  - Formatul urmeaza sa fie definit (nu sunt toti laboranzii definiti)
  - Nota laborator + bonus maxim 1p de la laborant (bonusul îl pot primi doar cei care au punctaj din teme)

Curs live/online + schimbari seminar

- Eu nu trebuia sa tin cursul semestrul asta , a aparut o modificare de ultim moment.... Prin urmare primele saptamani trebuie sa facem mici modificari in orar..
- Saptamana viitoare 22 februarie face curs online de la 16 la 18, s-ar putea sa se mai intample asta la 1-3 cursuri.
- Grupa 134 Seminar Joi 16-18 saptamana asta ? + peste 2 saptamani
- La grupa 132 (poate 134) o sa vina altcineva la primele 2 seminarii

Overview al materiei

- Curs 1-2 Sortări/Căutare binară
  - count sort, radix sort, quick sort, merge sort
- Curs 3 Vectori/Liste înlănțuite
  - Cozi
  - Stive
  - Deque
- Curs 4 Heapuri
- Curs 5 Heapuri binomiale - fibonacci
- Curs 6 Huffman
- Curs 7 Arbori binari de căutare
- Curs 8 AVL / Red black
- Curs 9 Skip Lists / Treaps
- Curs 10 Arbori de intervale
- Curs 11 RMQ & LCA & LA
- Curs 12-13 Hashuri
- Curs 14 Tries / Suffix trees ?

Overview al materiei

- Curs 1-2 Sortări/Căutare binară
  - count sort, radix sort, quick sort, merge sort
- Curs 3 Vectori/Liste înlănțuite
  - Cozi
  - Stive
  - Deque
- Curs 4 Heapuri
- Curs 5 Heapuri binomiale - fibonacci
- Curs 6 Huffman
- Curs 7 Arbori binari de căutare
- Curs 8 AVL / Red black
- Curs 9 Skip Lists / Treaps
- Curs 10 Arbori de intervale
- Curs 11 RMQ & LCA & LA
- Curs 12-13 Hashuri
- Curs 14 Tries / Suffix trees ?

Propuneri ?

-

Algoritmi de sortare

Ce algoritmi de sortare cunoașteți?

Algoritmi de sortare

Ce algoritmi de sortare cunoașteți?

- Bubble O(n^2)
- Merge O(nlogn)
- Interschimbare O(n^2)
- Radix
- Quick O(nlogn)?
- Heap O(nlogn)
- Bucket Sort
- Count Sort
- Bogo Sort O(n!*n)
- Gravity Sort O(n^2)
- Selection Sort O(n^2)
- Insert sort O(n^2)
- Shell Sort O(n sqrt n) ~ discutabil
- Intro Sort O(nlogn) alg hibrid
- Tim Sort O(nlogn) alg hibrid

Putem grupa după:

- Complexitate
- Complexitate spațiu
- Stabilitate
- Dacă se bazează pe comparații sau nu

Algoritmi de sortare stabili

- Un algoritm de sortare este stabil dacă păstrează ordinea elementelor egale.
- 5 5 5 -> 5 5 5 (sortare stabilă)
- 5 5 5 -> 5 5 5 (sortare instabilă) sau oricare alta permutare

Atenție: Și unii algoritmi instabili pot sorta stabil uneori, algoritmii stabili garantează asta pentru orice input.

Pentru numere naturale nu este important, dar când sortăm altfel de obiecte acest lucru poate deveni important.

Algoritmi de sortare

Clasificare

Elementari	Prin comparație	Prin numărare
Insertion sort → $O(n^2)$	Quick sort → O(n logn)	Bucket sort
Selection sort → $O(n^2)$	Merge sort → O(n logn)	Counting sort
Bubble sort → $O(n^2)$	Heap sort → O(n logn)	Radix sort
	Intro sort → O(n logn)	

Tabel cu sortări:
https://en.wikipedia.org/wiki/Sorting_algorithm#Comparison_of_algorithms

Sortare prin numărare / Counting Sort

- Algoritm de sortare a numerelor întregi mici
- Presupunem că vectorul de sortat **v** conține **n** elemente din mulțimea {0, ..., max}

IDEE:

- Creem un vector de frecvență **fr**
- Numărăm aparițiile fiecărui element din **v**
- Modificăm vectorul **fr** a.î.
  - `fr[i] = numărul de elemente cu valoare = i`
- La final, iterăm prin vectorul `fr[i]` și afișăm `i` de `fr[i]` ori pentru toate numerele de la 1 la max.

Sortare prin numărare / Counting Sort

Exemplu: sortăm note

Note	5	3	2	9	4	5	1	7	10	3	9	2	5
nota	1	2	3	4	5	6	7	8	9	10			
fr	1	2	2	1	3		1		2	1			

Sortare prin numărare / Counting Sort

Exemplu: sortăm note

Note	5	3	2	9	4	5	1	7	10	3	9	2	5
nota	1	2	3	4	5	6	7	8	9	10			
fr	1	2	2	1	3	0	1	0	2	1			

Sortare prin numărare / Counting Sort

Exemplu: sortăm note

nota	1	2	3	4	5	6	7	8	9	10			
fr	1	2	2	1	3	0	1	0	2	1			
soluție	1	2	2	3	3	4	5	5	5	7	9	9	10

Sortare prin numărare / Counting Sort

Exemplu: sortăm note

Note	5	3	2	9	4	5	1	7	10	3	9	2	5
soluție	1	2	2	3	3	4	5	5	5	7	9	9	10

Cod

```
//Pasul 1: Crestem frecventa fiecarui element din vector:

 for (int i = 0; i < n; ++i)   // O(n)
        fr[note[i]]++, maxn = max(maxn, note[i]);

// Pasul 2: afisam fiecare element de atatea ori cat apare in vectorul de frecventa
// O(maxn * n)
// O(maxn + n)

 for (int i = 0; i <= maxn; ++i) {    // pana la maxn
        for (int j = 1; j <= fr[i]; ++j) {   // worst case se duce pana la n
                                             // for-ul va face n + maxn
            cout << i << " ";  // Afisam de fix n ori
        }
 }
```

Complexitate? Spațiu? Timp?

Counting Sort

Complexitate

- Timp:
  - O(n + max)

- Spațiu:
  - O(max)

Counting Sort

Vizualizare:

https://visualgo.net/bn/sorting

Counting Sort

Ce ne facem dacă avem de sortat numere mari...

- Până la 10^6 ?
- Până la 10^{18} ?
- Numere care nu sunt întregi ?

Counting Sort

Ce ne facem dacă avem de sortat numere mari...

- Până la 10^6 ?
  - Depinde de N, dar **Count Sort** poate fi cea mai bună opțiune...
- Până la 10^{18} ?
  - Nu mai putem folosi Count Sort. Putem folosi în schimb **Radix Sort**
- Numere care nu sunt întregi ?
  - Mai greu și cu Radix Sort (nu e imposibil, dacă sunt doar 1-2 zecimale putem înmulți cu 10, 100) ... altfel putem folosi **Bucket Sort**

Kahoot

https://create.kahoot.it/creator/2281f10b-f400-43fe-981c-85377fd66c12

Final