TUTORIAT -5- P. A.

COMPLEXITATEA ALGORITMILOR

Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)

PENTRU FIECARE DIN URMATOARELE PROBLEME, SPECIFICATI SI COMPLEXITATEA TIMP (EVENTUAL SI SPATIU)

1. AVAND CA INPUT O SITUATIE DE JOC X SI 0, SCRIETI UN PROGRAM CARE DECIDE

DACA/CINE A CASTIGAT. EXEMPLU: 0 REPREZINTA SPATII GOALE

[1,2,0]

[2,1,0] - 1 A CASTIGAT

[2,0,1]

2. AVAND DOUA LISTE ORDONATE CRESCATOR, INTERCLASATI-LE INTR-O LISTA ORDONATA TOT CRESCATOR.

EXEMPLU:

A: 1,2,5,7,8

B: 0,3,5,9,10

RASPUNS: 0,1,2,3,5,5,7,8,9,10

3. COWS AND BULLS SE JOACA ASTFEL: SE GENEREAZA UN NUMAR RANDOM DE 4 CIFRE. JUCATORUL INCEARCA SA IL GHICEASCA LA INTAMPLARE SI PRIMESTE INAPOI NUMARUL DE CIFRE GHICITE PE LOCUL POTRIVIT SI NR DE CIFRE GHICITE PE LOCUL GRESIT. EXEMPLU:

NUMARUL ESTE 1038.

JUCATORUL INTRODUCE: 1234

- 2 COWS, 0 BULLS (1 SI 3 SUNT GHICITE PE LOCURILE BUNE, NU EXISTA

O ALTA CIFRA GHICITA DAR PE UN LOC GRESIT)

JUCATORUL INTRODUCE: 1286

- 1 COW, 1 BULL (1 E PE POZITIA BUNA, 8 ESTE IN NUMARUL CORECT DAR NU ESTE PE POZITIA BUNA)

IMPLEMENTATI JOCUL.

4. DETERMINATI VARFUL GRAFICULUI PENTRU O ECUATIE DE GRADUL DOI.

$$y = a x^2 + b x + c$$

5. Scrieti un program care determina cel mai mare divizor comun al unei liste de numere intregi.

Exemplu:

```
GCD([1024, 192, 2048, 512]) -> 64
GCD([10, 20, 40]) → 10
```

6. Scrieti o functie care primeste n numar intreg si afiseaza matricea identitate de marime n x n.

7. Scrieti un program care calculeaza cate cifre sunt intre 0 si o valoare primita ca input.

Exemplu : digits(10) = 9

DIGITS(100) = 189

8. INSERTION SORT

9. Count Sort

10. Cautare binara pe un sir ordonat crescator

11. Se citesc n intervale închise. Să se calculeze suma lungimilor lor.

12. Scrieti o functie care afișează puterile lui 2 cuprinse între a și b

13. Scrieti o functie care sorteaza partial un sir, adaugand numerele negative la inceputul lui.