cs223fig2.cdr
Tuesday, January 25, 2022 11:30:43 AM
Color profile: Disabled

Composite Default screen

Algorithm Efficiency

📖 Chapter 2

➪ **Space complexity**

➪ **Time complexity**

➪ How well it performs?

- ✎ Space efficiency
- ▶ ✎ Time efficiency

➪ A framework for analysis

A note on homework exercises and weekly tutorial sessions.

➪ Approach to chapter (caution)

© 2022 Dr. Muhammad Al-Hashimi

cs223fig2.cdr
Tuesday, January 25, 2022 11:30:43 AM
Color profile: Disabled

Composite Default screen

Analysis Framework

- Input size as parameter
- Measuring run time
- Run time growth
- Algorithm efficiency
- Types of efficiency analysis

© 2022 Dr. Muhammad Al-Hashimi

cs223fig2.cdr
Tuesday, January 25, 2022 11:30:43 AM
Color profile: Disabled

Composite Default screen

Orders of Growth

These functions arise naturally from the way we design algorithms.

How fast *f* grows as *n* increases?

TABLE 2.1 Values (some approximate) of several functions important for analysis of algorithms

n	$\log_2 n$	n	$n \log_2 n$	n^2	n^3	2^n	$n!$
10	3.3	10^1	$3.3 \cdot 10^1$	10^2	10^3	10^3	$3.6 \cdot 10^6$
10^2	6.6	10^2	$6.6 \cdot 10^2$	10^4	10^6	$1.3 \cdot 10^{30}$	$9.3 \cdot 10^{157}$
▶ 10^3	10	10^3	$1.0 \cdot 10^4$	10^6	10^9		
10^4	13	10^4	$1.3 \cdot 10^5$	10^8	10^{12}		
▶ 10^5	17	10^5	$1.7 \cdot 10^6$	10^{10}	10^{15}		
10^6	20	10^6	$2.0 \cdot 10^7$	10^{12}	10^{18}		

Exercise
Determine resulting growth when n doubles (increases twofold) for each function?

© 2022 Dr. Muhammad Al-Hashimi

cs223fig2.cdr
Tuesday, January 25, 2022 11:30:43 AM
Color profile: Disabled

Composite Default screen

Analysis of Algorithms
Types of Efficiency

- **Worst-case efficiency**
- **Average-case efficiency**

Runtime of a *selection sort* always depends on list size only, is this true for the search?

Runtime growth can vary for the same input size.

Algorithm *SequentialSearch*
Input Array $A[0\,..\,n-1]$, saerch key K
Output Index of first element of A to match K, otherwise -1

1: $i \leftarrow 0$
2: **while** $i < n$ **and** $A[i] \neq K$ **do**
3: $\quad i \leftarrow i + 1$
4: **if** $i < n$ **then return** i ▷ found if i in range
5: **return** -1

© 2022 Dr. Muhammad Al-Hashimi

cs223fig2.cdr
Tuesday, January 25, 2022 11:30:43 AM
Color profile: Disabled

Composite Default screen

Analysis of Algorithms
A General Plan

🕐 ➪ Amortized efficiency

1. Select suitable input size parameter
2. Identify suitable basic operation
3. Check dependence of basic op
4. Determine count $C(n)$, somehow
5. Determine order of growth of $C(n)$

🗝 **Some algorithms have different counts for the same input size.**

🗝 **Run time grows similar to growth of basic operation count.**

© 2022 Dr. Muhammad Al-Hashimi