Strukture podataka i algoritmi

Peto predavanje

Funkcije rasta algoritama
Asimptotska notacija

Sortiranje umetanjem listi potprogram

- MERGE(A, p, q, r)

1. n1 ← q - p +1
2. n2 ← r - q
3. napravi polja L[1,.. n1+1], R[1,.. n2+1]
4. za i ← 1 do n1
5. L [i] ← A [p + i - 1]
6. za j ← 1 do n2
7. R [j] ← A [q + j]
8. L [n+1] ← ∝
9. R [n+1] ← ∝
10. i ← 1
11. j ← 1

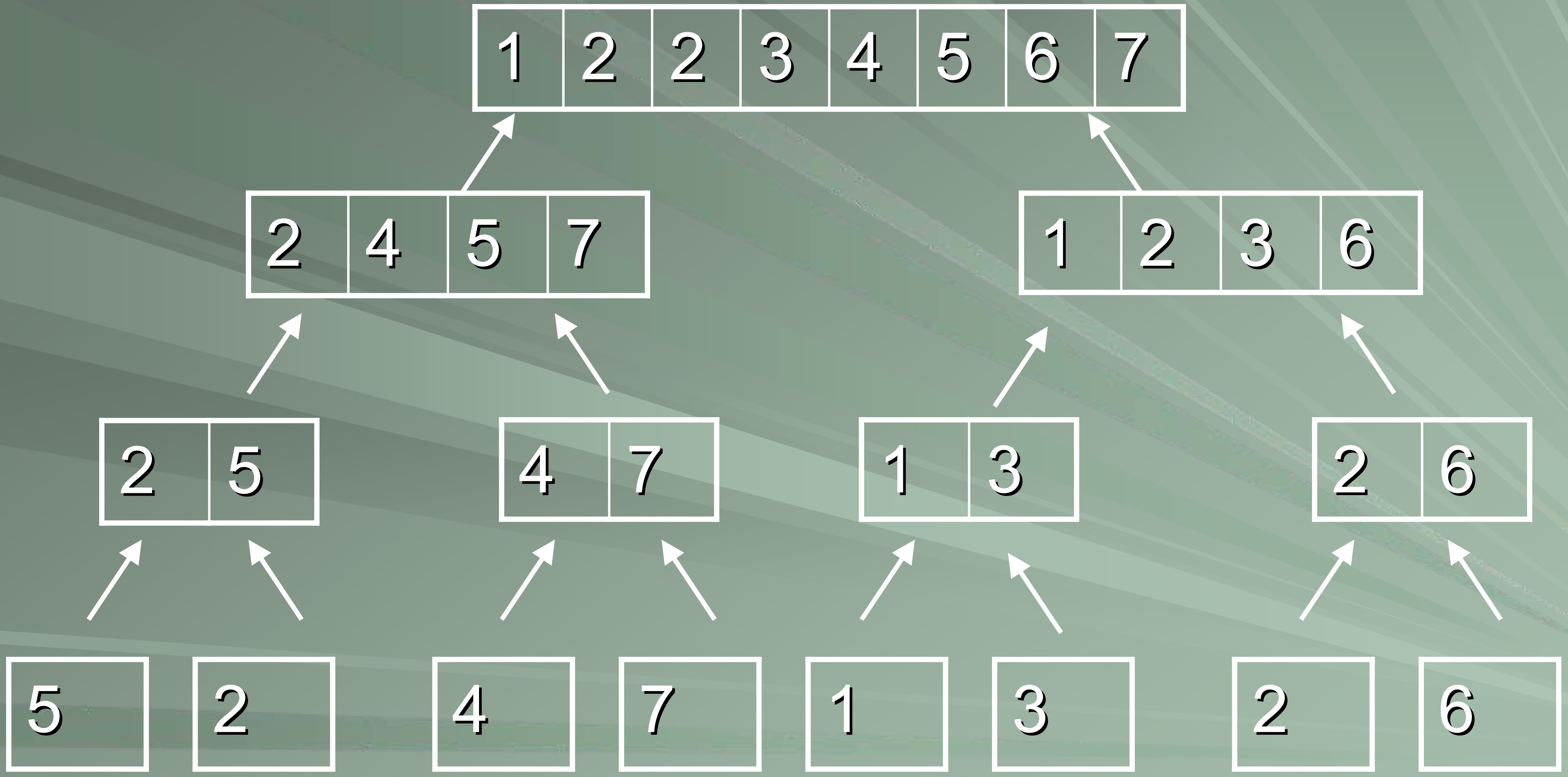
10) za k ← p do r
11) ako je L[i] ≤ R[j]
12) tada A [k] ← L [i]
13) i ← i + 1
14) inače A [k] ← R [j]
15) j ← j + 1

Sortiranje umetanjem listi glavni program

- MERGE(A, p, r)

1. If p<r
2. tada q ← [(p+r)/2]
3. MERGE-SORT (A,p,q)
4. MERGE-SORT (A,q+1,r)
5. MERGE(A,p,q,r)

Sekvenca sortiranja



Analiza utrošenog vremena

- Direktno rješenje $\theta(1)$
- Problem ima ***a*** pod problema od kojih je svaki ***1/b*** veličino od problema
- Vrijeme izvršenja
- $$T(n) = \begin{cases} \theta(1) & \text{za } n<c \\ aT(n/b)+D(n)+C(n) & \text{ostalo} \end{cases}$$
- C(n) – vrijeme potrebno za podjelu problema
- D(n) – vrijeme potrebno za združivanje rješenja

Naš slučaj

- a=2
- b=2
- D(n)= $\theta(1)$
- T(n)= $\begin{cases} \theta(1) & \text{za } n=1 \\ 2T(n/2)+\theta(n) & \text{za } n>1 \end{cases}$

Odnosno

- T(n)= $\begin{cases} c & \text{za } n=1 \\ 2T(n/2)+cn & \text{za } n>1 \end{cases}$

Utrošeno vrijeme -- T(n)

>ukupno cn *lg*n +cn koraka

Funkcija rasta

$\theta(n\ \mathit{ln}\ n)$

Miješanje algoritamskih pristupa

- Ograničeni pod problem (algoritam ne mora imati najpovoljniju funkciju rasta
- Razbijanje problema na pod probleme koji se rješavaju različitim algoritmima

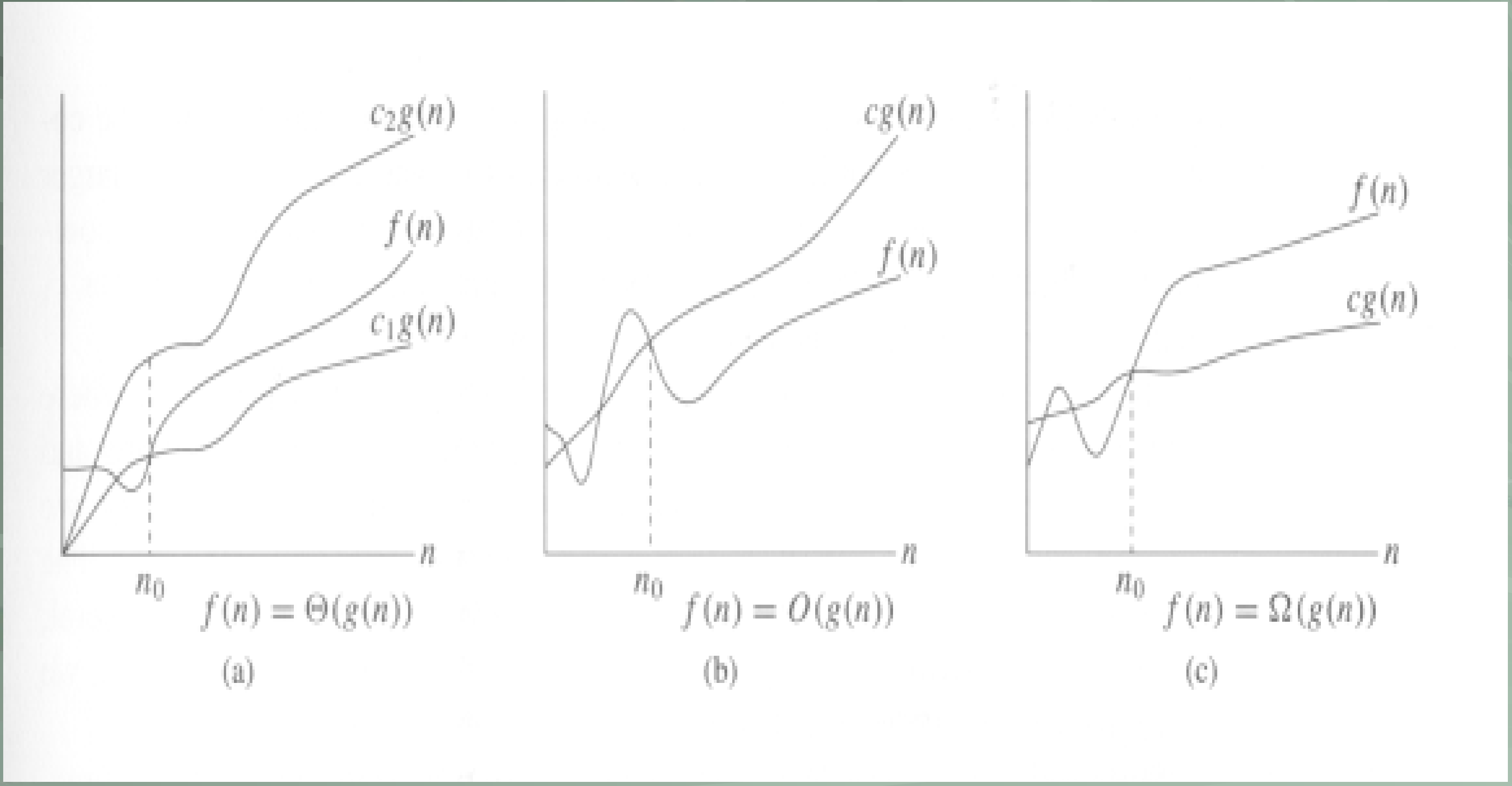
Bublesort

1) za i← 1 do dužina [A]
2) za j← dužina[A] do i+1
3) ako A[j]< A[j-1]
4) izmjeni A[j]← →A[j-1]

Funkcije rasta

- Asimptotski karakter
- Predstavljaju skup ili klasu funkcija koje zadovoljavanju neke (zadane) uvjete
- Vrijeme izvršenja algoritma (bez obzira na na povoljnost slučaja pripada zadanoj klasi
- Asiptotska notacija
  - Θ (g(n))
  - Ω (g(n))
  - O (g(n))

Primjeri


$c_2g(n)$
$f(n)$
$c_1g(n)$
n
n_0
$f(n) = \Theta(g(n))$
(a)
$cg(n)$
$f(n)$
n
n_0
$f(n) = O(g(n))$
(b)
$f(n)$
$cg(n)$
n
n_0
$f(n) = \Omega(g(n))$
(c)

Θ (g(n))

- Θ (g(n)) = {f(n) : tada postoje pozitivne konstante c_1,c_2 i n_0 takove da
  - $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ za sve $n \ge n_0$ }
- Θ (g(n)) je skup (klasa) svih funkcija f(n) koje zadovoljavaju uvjet

Ω (h(n)) asimptotska donja granica

- Ω(g(n)) = {f(n) : tada postoje pozitivne konstante c i n_0 takove da
  - $0 \leq cg(n) \leq f(n)$ za sve $n \geq n_0$ }

O(g(n)) asimptotska gornja granica

- O(g(n)) = {f(n) : tada postoje pozitivne konstante c i n_0 takove da
  - 0 ≤f(n) ≤ cg(n) za sve n≥ n_0

Zadaća

	1 sec	1 min	1 sat	1 dan	1 mj	1 god	1 st
lg n							
$\sqrt{n}$							
n							
n *lg* n							
n^2							
n^3							
2^n							
n!							

Za svaku funkciju *f(n)* i vrijeme *t* u tablici izračunajte najveći
n problema koji može biti riješen u tom vremenu uz pretpostavku
da algoritam koji rješava problem treba f(n) mikrosekundi.