

Objectives

- Divide and conquer algorithms
 - Counting inversions
 - Closest pair of points

Mar 4, 2013

CSCI211 - Sprenkle

1

Review

- What is a recurrence relation?
- How can you compute D&C running times?
 - 2 ways to solve

Mar 4, 2013

CSCI211 - Sprenkle

2

COUNTING INVERSIONS

Mar 4, 2013

CSCI211 - Sprenkle

3

Comparing Rankings

- To determine similarity of rankings, need a metric
- **Similarity metric:** number of inversions between two rankings
 - My rank: 1, 2, ..., n
 - Your rank: a_1, a_2, \dots, a_n
 - Movies i and j *inverted* if $i < j$ but $a_i > a_j$

Naïve/Brute force solution?

	Movies				
	A	B	C	D	E
Me	1	2	3	4	5
You	1	3	4	2	5

Inversions:
3-2, 4-2

Mar 4, 2013

CSCI211 - Sprenkle

4

Brute Force Solution

- Look at every pair (i, j) and determine if they are an inversion
- Requires $\Theta(n^2)$ time
 - Note: Already an efficient algorithm but try to improve upon runtime

Towards a Better Solution...

- Can't look at each inversion individually

Mar 4, 2013

CSCI211 - Sprenkle

5

Counting Inversions: Divide-and-Conquer

Assume number represents where item *should* be in the list, i.e., where it is in someone else's list

1 5 4 8 10 2 6 9 12 11 3 7

↑
Should be at position 5

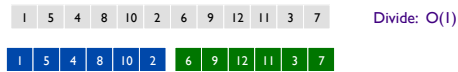
Mar 4, 2013

CSCI211 - Sprenkle

6

Counting Inversions: Divide-and-Conquer

- **Divide:** separate list into two pieces



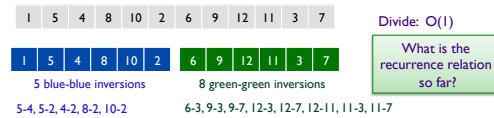
Mar 4, 2013

CSCI211 - Sprenkle

7

Counting Inversions: Divide-and-Conquer

- **Divide:** separate list into two pieces
- **Conquer:** recursively count inversions in each half



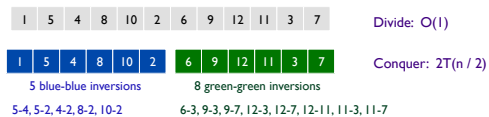
Mar 4, 2013

CSCI211 - Sprenkle

8

Counting Inversions: Divide-and-Conquer

- **Divide:** separate list into two pieces
- **Conquer:** recursively count inversions in each half



What do we need to do next?

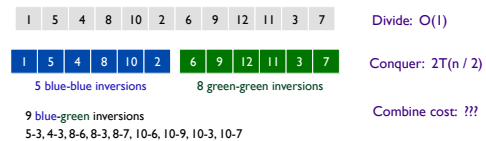
Mar 4, 2013

CSCI211 - Sprenkle

9

Counting Inversions: Divide-and-Conquer

- **Divide:** separate list into two pieces
- **Conquer:** recursively count inversions in each half
- **Combine:** count inversions where a_i and a_j are in different halves, and return sum of three quantities



Total = 5 + 8 + 9 = 22

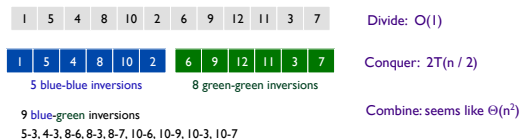
Mar 4, 2013

CSCI211 - Sprenkle

10

Counting Inversions: Divide-and-Conquer

- **Divide:** separate list into two pieces
- **Conquer:** recursively count inversions in each half
- **Combine:** count inversions where a_i and a_j are in different halves, and return sum of three quantities



Total = 5 + 8 + 9 = 22

What would make figuring out blue-green inversions easier?

Mar 4, 2013

CSCI211 - Sprenkle

11

Counting Inversions: Combine

Combine: count blue-green inversions

- Assume each half is sorted
 - Count inversions where a_i and a_j are in different halves
 - Merge two sorted halves into sorted whole
- to maintain sorted invariant



- What does sorting do for us?
- What is our algorithm for counting the inversions and merging?

Mar 4, 2013

CSCI211 - Sprenkle

12

Counting Inversions: Combine

Combine: count blue-green inversions

- Assume each half is sorted
- Count inversions where a_i and a_j are in different halves
- Merge two sorted halves into sorted whole

to maintain sorted invariant

Count: $O(n)$

13 blue-green inversions: $6 + 3 + 2 + 2 + 0 + 0$

Merge: $O(n)$

We'll run through an example in a bit...

Mar 4, 2013

CSCI211 - Sprenkle

13

Counting Inversions: Implementation

```
Sort-and-Count(L)
  if list L has one element
    return 0 and the list L

  Divide the list into two halves A and B
  (iA, A) = Sort-and-Count(A)
  (iB, B) = Sort-and-Count(B)
  (i, L) = Merge-and-Count(A, B)

  total_inversions = iA + iB + i
  return total_inversions and the sorted list L
```

Mar 4, 2013

CSCI211 - Sprenkle

14

Counting Inversions: Implementation

```
Sort-and-Count(L)
  if list L has one element
    return 0 and the list L

  Divide the list into two halves A and B
  (iA, A) = Sort-and-Count(A)
  (iB, B) = Sort-and-Count(B)
  (i, L) = Merge-and-Count(A, B)

  total_inversions = iA + iB + i
  return total_inversions and the sorted list L
```

- Merge-and-Count
 - Pre-condition. A and B are sorted.
 - Post-condition. L is sorted.

Mar 4, 2013

CSCI211 - Sprenkle

15

Merge and Count

```
Merge-and-Count(A,B):
  i=0
  j=0
  inversions = 0
  output = []
  while i < A.size and j < B.size:
    output.append( min(A[i], B[j]) )
    if B[j] < A[i]:
      inversions += A.size - i
    update i or j
  Append the remainder of the non-exhausted list to
  the output
  return inversions and output
```

Mar 4, 2013

CSCI211 - Sprenkle

16

Merge and Count

Precondition: A and B are sorted

```
Merge-and-Count(A,B):
  i=0 (front of list A)
  j=0 (front of list B)
  inversions = 0
  output = []
  while A not empty and B not empty:
    output.append( min(A[i], B[j]) )
    if B[j] < A[i]:
      inversions += A.size - i (remaining elements in A)
    update i or j (whichever had smaller element)
  Append the remainder of the non-exhausted list to
  the output
  return inversions and output
```

Mar 4, 2013

CSCI211 - Sprenkle

17

Merge and Count Step

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole

↓ ↓ two sorted halves

A 3 7 10 14 18 19 B 2 11 16 17 23 25

Output array

Total:

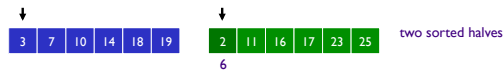
Mar 4, 2013

CSCI211 - Sprenkle

18

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 Output array

Total: 6

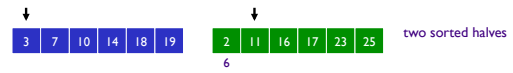
Mar 4, 2013

CSCI211 - Sprenkle

19

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 Output array

Total: 6

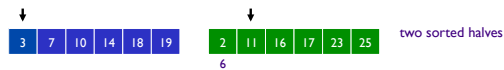
Mar 4, 2013

CSCI211 - Sprenkle

20

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 3 Output array

Total: 6

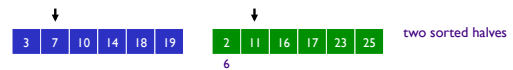
Mar 4, 2013

CSCI211 - Sprenkle

21

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 3 Output array

Total: 6

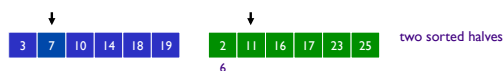
Mar 4, 2013

CSCI211 - Sprenkle

22

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 3 7 Output array

Total: 6

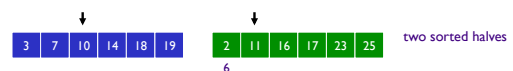
Mar 4, 2013

CSCI211 - Sprenkle

23

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



2 3 7 Output array

Total: 6

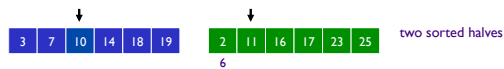
Mar 4, 2013

CSCI211 - Sprenkle

24

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10

Total: 6

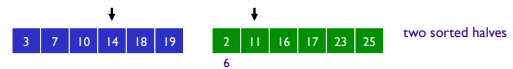
Mar 4, 2013

CSCI211 - Sprenkle

25

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10

Total: 6

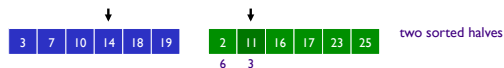
Mar 4, 2013

CSCI211 - Sprenkle

26

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10 11

Total: 6 + 3

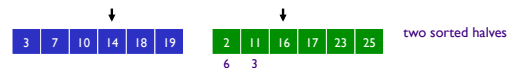
Mar 4, 2013

CSCI211 - Sprenkle

27

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10 11

Total: 6 + 3

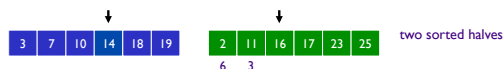
Mar 4, 2013

CSCI211 - Sprenkle

28

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10 11 14

Total: 6 + 3

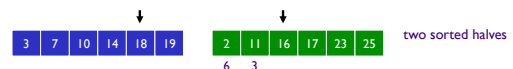
Mar 4, 2013

CSCI211 - Sprenkle

29

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Output array: 2 3 7 10 11 14

Total: 6 + 3

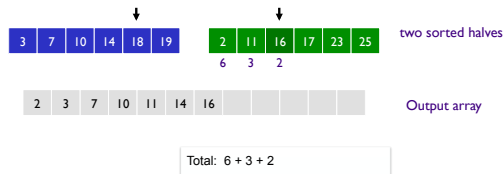
Mar 4, 2013

CSCI211 - Sprenkle

30

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



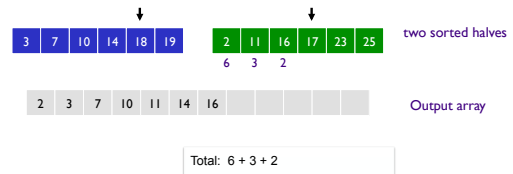
Mar 4, 2013

CSCI211 - Sprenkle

31

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



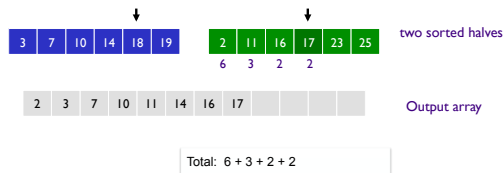
Mar 4, 2013

CSCI211 - Sprenkle

32

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



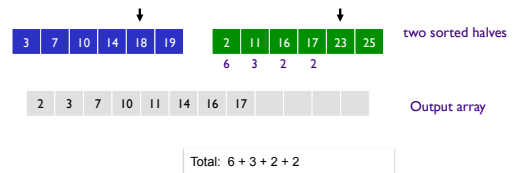
Mar 4, 2013

CSCI211 - Sprenkle

33

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



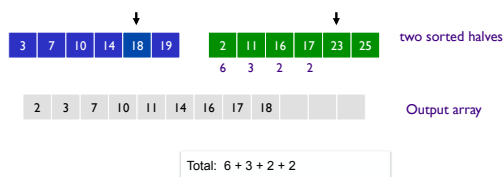
Mar 4, 2013

CSCI211 - Sprenkle

34

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



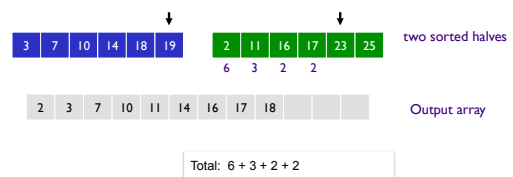
Mar 4, 2013

CSCI211 - Sprenkle

35

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



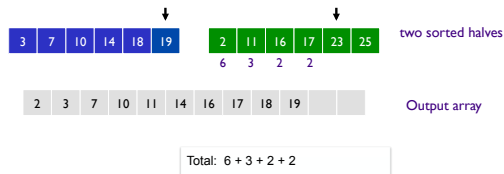
Mar 4, 2013

CSCI211 - Sprenkle

36

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



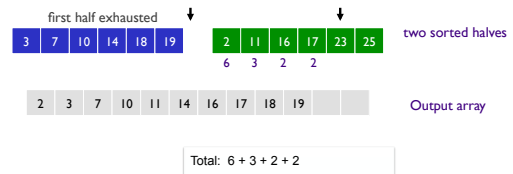
Mar 4, 2013

CSCI211 - Sprenkle

37

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



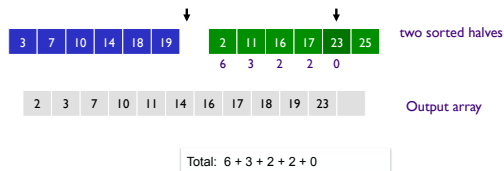
Mar 4, 2013

CSCI211 - Sprenkle

38

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



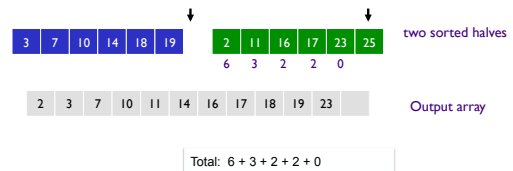
Mar 4, 2013

CSCI211 - Sprenkle

39

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



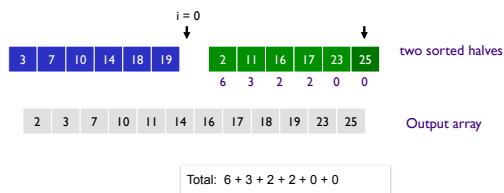
Mar 4, 2013

CSCI211 - Sprenkle

40

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



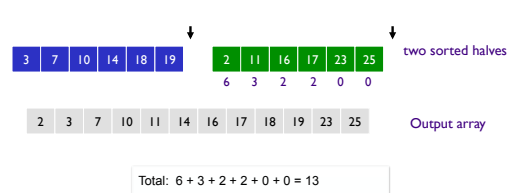
Mar 4, 2013

CSCI211 - Sprenkle

41

Merge and Count

- Given two sorted halves, count number of inversions where a_i and a_j are in different halves
- Combine two sorted halves into sorted whole



Mar 4, 2013

CSCI211 - Sprenkle

42

Counting Inversions: Implementation

```

Sort-and-Count(L)
  if list L has one element
    return 0 and the list L

  Divide the list into two halves A and B
  (iA, A) = Sort-and-Count(A)
  (iB, B) = Sort-and-Count(B)
  (i, L) = Merge-and-Count(A, B)

  total_inversions = iA + iB + i
  return total_inversions and the sorted list L

```

Recurrence relation?
Runtime of algorithm?

- Merge-and-Count
 - Pre-condition. A and B are sorted.
 - Post-condition. L is sorted.

Mar 4, 2013

CSCI211 - Sprenkle

43

Analysis

Recurrence Relation:

$$T(n) \leq T(n/2) + T(n/2) + O(n)$$

$$\rightarrow T(n) \in O(n \log n)$$

```

Sort-and-Count(L)
  if list L has one element
    return 0 and the list L

  Divide the list into two halves A and B
  (iA, A) = Sort-and-Count(A)    T(n/2)
  (iB, B) = Sort-and-Count(B)    T(n/2)
  (i, L) = Merge-and-Count(A, B) O(n)

  return i = iA + iB + i and the sorted list L

```

Mar 4, 2013

CSCI211 - Sprenkle

44

Looking Ahead

- Wiki: 4.8, 5.1, 5.2 - Tuesday
- Problem Set 6 – due Friday – SSA day

Mar 4, 2013

CSCI211 - Sprenkle

45