BLG 336E
Analysis of Algorithms II

Practice Session 3

04.April.17

Dilara Torunoğlu-Selamet Istanbul Technical University – Department of Computer Engineering

Outline

- Divide and Conquer Algorithm
  - Introduction
  - Solve a Recurrence
  - Counting Inversion Example

Introduction

- Divide and Conquer is a different framework. Related to induction:
  - Suppose you have a "box" that can solve problems of size$\leq k < n$
  - You use this box on some subset of the input items to get partial answers.
  - You combine these partial answers to get the full answer.

  But: you construct the "box" by recursively applying the same idea until the problem is small enough to be solved by brute force.

Introduction

```
function mergesort(a[1...n])
Input:  An array of numbers a[1...n]
Output:  A sorted version of this array

if n > 1:
  return merge(mergesort(a[1...⌊n/2⌋]), mergesort(a[⌊n/2⌋+1...n]))
else:
  return a
```

- In practice, you sort in-place rather than making new lists.
- Total time: $T(n) \leq 2T(n/2) + c(n)$.

Introduction

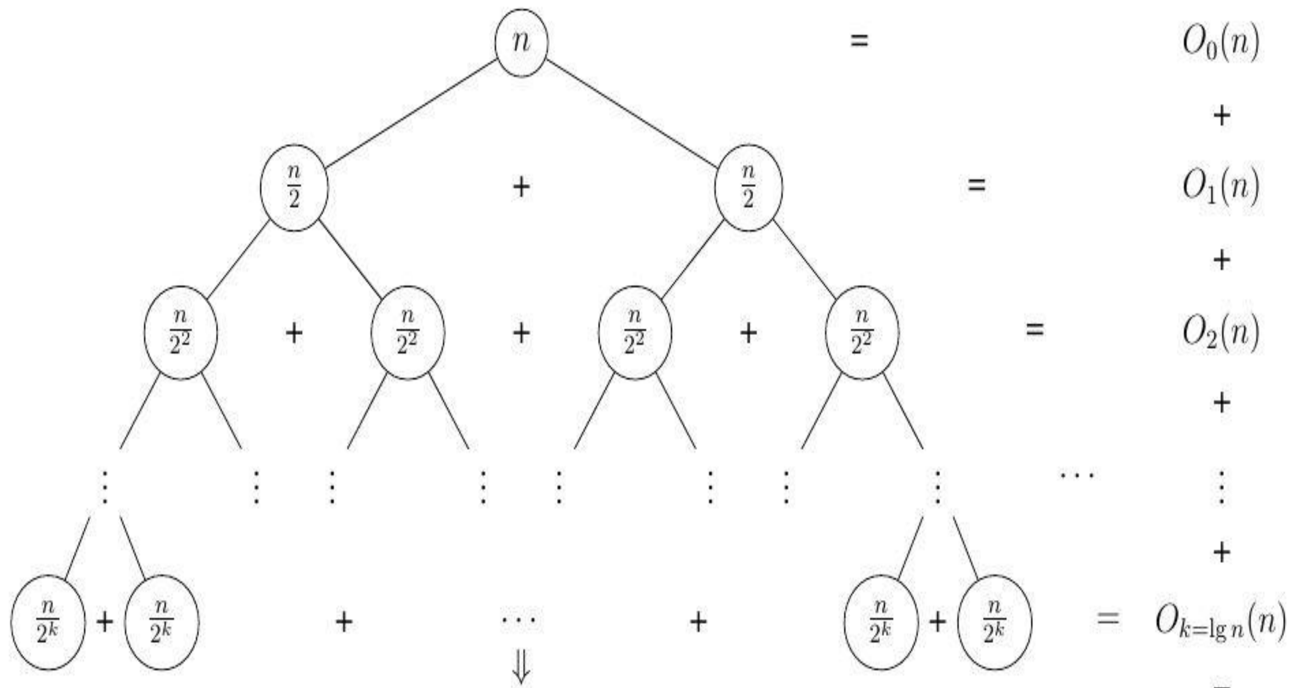
- How do you compute the time complexity of a divide and conquer algorithm for a certain input size?
    - $T(n) = a * T(n/b) + c(n)$
- We need to consider the
    - how many sub problems need to be solved (a), mostly $n/2$
    - what is the size of the sub problems (n/b)
    - cost of combination $c(n)$

Solve a Recurrence

- Given a recurrence such as
  - $T(n) \leq 2T(n/2) + c(n)$,
  - we want a simple upper bound on the total running time.
- Three common ways to solve such a recurrence:
  - Unroll the recurrence and see what the pattern is:
    - Draw the recursion tree
    - Substation method
    - Master Theorem

Solve a Recurrence

$$T(n) = 2(n/2) + O(n)$$



$$O\left(\sum_{i=0}^{k} 2^i \cdot \frac{n}{2^i}\right) = O\left(\sum_{i=0}^{k} n\right) = O(k \cdot n) \Leftrightarrow O(n \cdot \log n)$$

Dilara Torunoğlu-Selamet Istanbul Technical University – Department of Computer Engineering

Solve a Recurrence

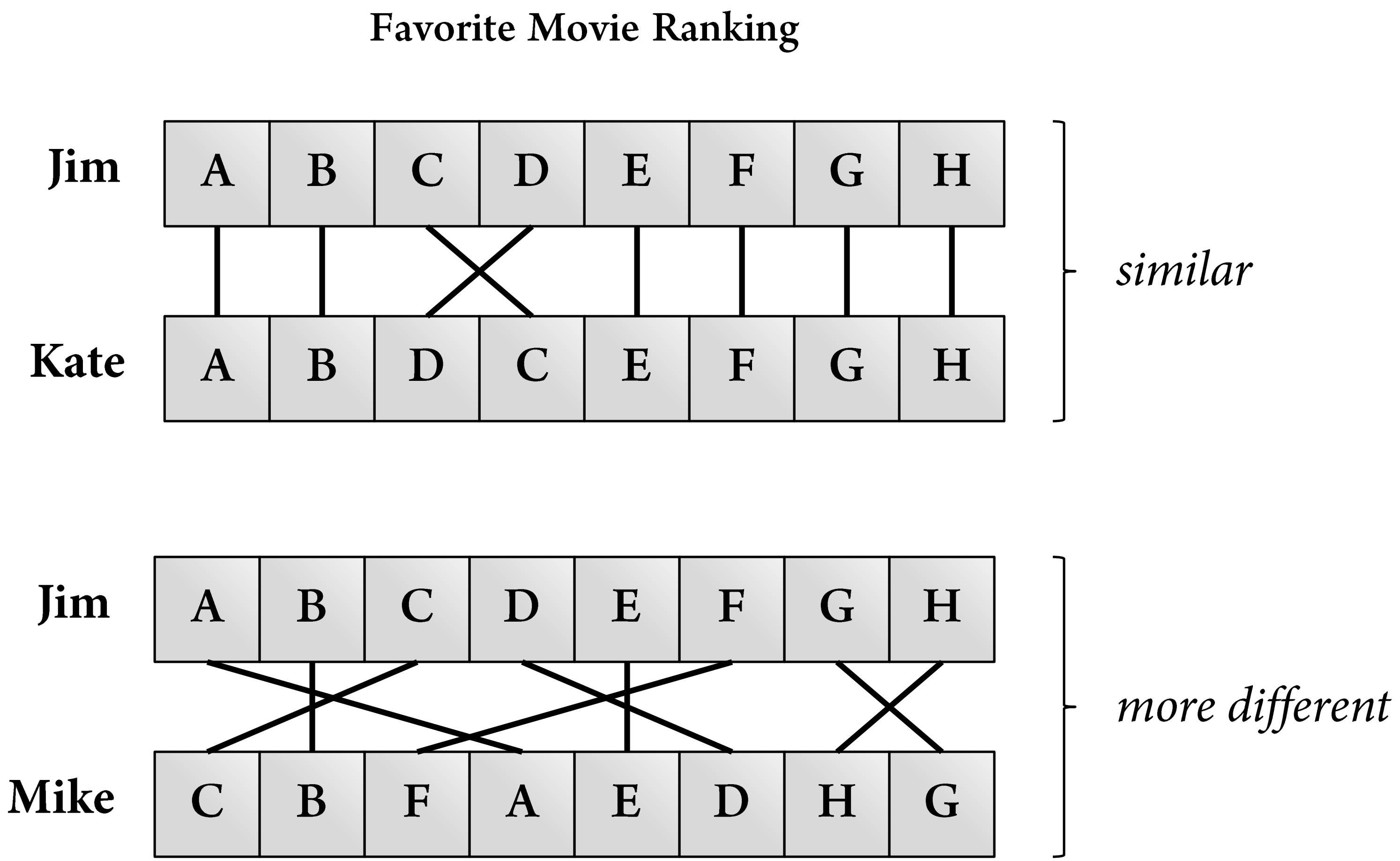
- The root node represents the original problem.
- Every node that is not a leaf has *a* children, representing the number of sub problems it is splitting into.
- Find the size of the sub problem with the help of *b*
- The work at the leaves is $T(1)$
- The work done at each level stays consistent (in this case, it is $O(n)$ at each level)

$$\underbrace{\boxed{2^i}}_{\text{number of subproblems}} * \underbrace{\boxed{\frac{n}{2^i}}}_{\text{size of the subproblem}} = O(n)$$

There are log(n) level. Hence
O(n*log(n))

Dilara Torunoğlu-Selamet Istanbul Technical University – Department of Computer Engineering

Counting Inversions



Counting Inversions

The items in the reference ranking are mapped to **incremental** indices, while those in the second ranking are numbered according to this mapping.

Any pair in the second ranking, where the element on the left is larger than the element on the right, is called an **inversion**.

The **number of inversions** in the second ranking is a **metric for its dissimilarity** to the reference ranking.

Jim	A	B	C	D	E	F	G	H
	1	2	3	4	5	6	7	8

Kate	A	B	D	C	E	F	G	H
	1	2	4	3	5	6	7	8

Counting Inversions

Brute Force

C	B	F	A	E	D	H	G
3	**2**	**6**	**1**	**5**	**4**	**8**	**7**
3↔2 3↔1	2↔1	6↔1 6↔5 6↔4		5↔4		8↔7	

Total number of inversions = **8**

Counting Inversions

Divide and Conquer

C	B	F	A	E	D	H	G
3	2	6	1	5	4	8	7

Total number of inversions = **??**

Counting Inversions

Divide and Conquer

C	B	F	A	E	D	H	G
3	2	6	1	5	4	8	7

Total number of inversions = **??**

RECURSION

Counting Inversions

Divide and Conquer

C	B	F	A
3	**2**	**6**	**1**

#inversions in this part = **??**

E	D	H	G
5	**4**	**8**	**7**

#inversions in this part = **??**

#inversions from merging = **??**

Counting Inversions

Divide and Conquer

C	B	F	A
3	**2**	**6**	**1**

#inversions in this part = **??**

RECURSION

E	D	H	G
5	**4**	**8**	**7**

#inversions in this part = **??**

#inversions from merging = **??**

Counting Inversions

Divide and Conquer

C	B
3	**2**

#invs = **??**

F	A
6	**1**

#invs = **??**

#inversions = **??**

E	D	H	G
5	**4**	**8**	**7**

Counting Inversions

Divide and Conquer

B	C
2	**3**

#invs = **1**

F	A
6	**1**

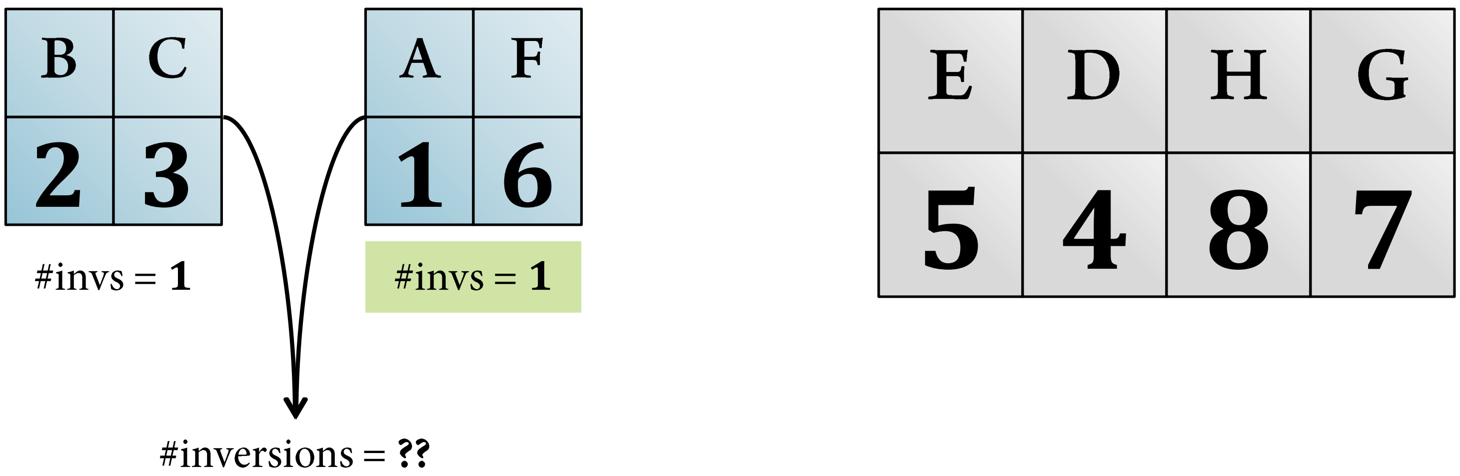
#invs = **??**

#inversions = **??**

E	D	H	G
5	**4**	**8**	**7**

Counting Inversions

Divide and Conquer



Counting Inversions

Divide and Conquer

B	C
2	**3**

#invs = **1**

A	F
1	**6**

#invs = **1**

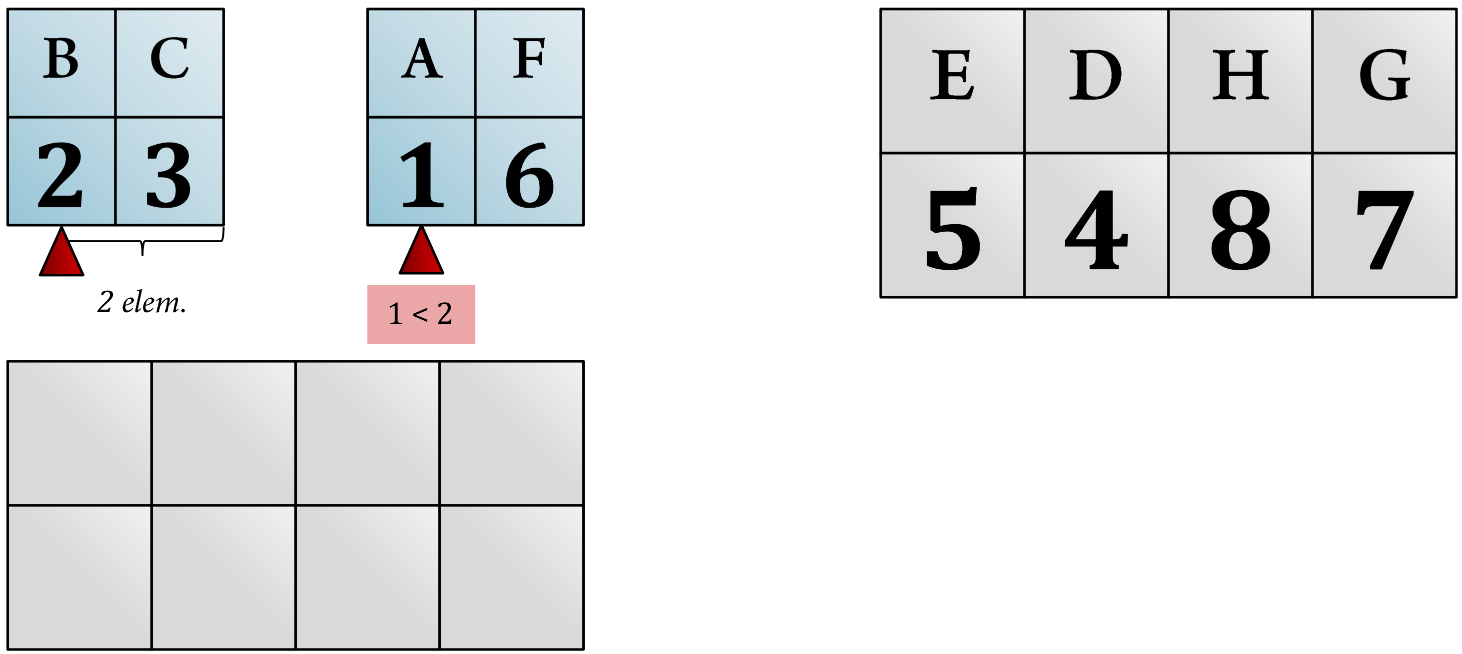
#inversions = **??**

MERGE AND COUNT

E	D	H	G
5	**4**	**8**	**7**

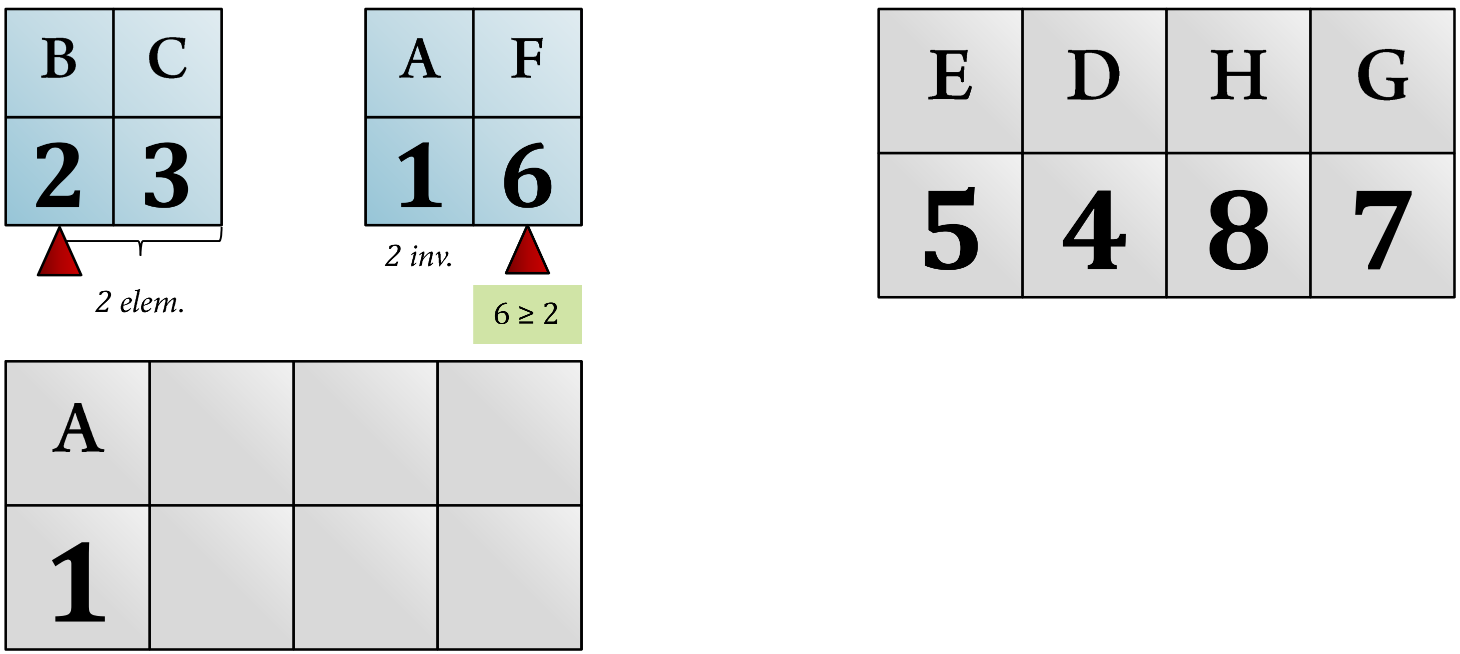
Counting Inversions

Divide and Conquer



Counting Inversions

Divide and Conquer



Counting Inversions

Divide and Conquer

B	C
2	3

1 elem.

A	F
1	6

2 inv.

6 ≥ 3

E	D	H	G
5	4	8	7

A	B		
1	2		

Counting Inversions

Divide and Conquer

B	C
2	3

A	F
1	6

2 inv.

E	D	H	G
5	4	8	7

A	B	C	
1	2	3	

Counting Inversions

Divide and Conquer

B	C
2	3

A	F
1	6
2 inv.	*0 inv.*

E	D	H	G
5	4	8	7

A	B	C	F
1	2	3	6

Counting Inversions

Divide and Conquer

B	C
2	**3**

#invs = **1**

A	F
1	**6**

#invs = **1**

#inversions = **2**

E	D	H	G
5	**4**	**8**	**7**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

#inversions in this part = **4**

E	D	H	G
5	**4**	**8**	**7**

#inversions in this part = **??**

#inversions from merging = **??**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

#inversions in this part = **4**

E	D	H	G
5	**4**	**8**	**7**

#inversions in this part = **??**

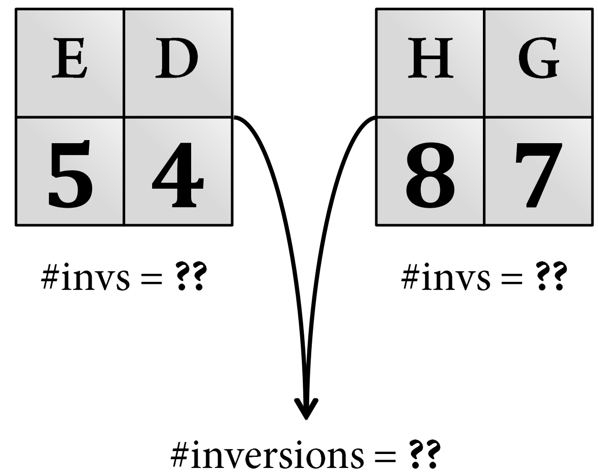
RECURSION

#inversions from merging = **??**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**



Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

#invs = **1**

H	G
8	**7**

#invs = **??**

#inversions = **??**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

#invs = **1**

G	H
7	**8**

#invs = **1**

#inversions = **??**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

#invs = **1**

G	H
7	**8**

#invs = **1**

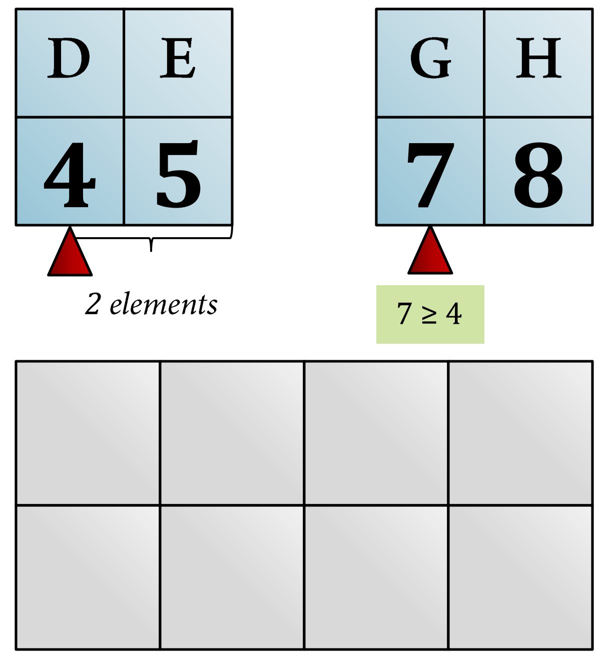
#inversions = **??**

MERGE AND COUNT

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6



Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

1 elem.

G	H
7	**8**

$7 \geq 5$

D			
4			

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

G	H
7	**8**

D	E		
4	**5**		

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E
4	5

G	H
7	8

0 inv.

D	E	G	
4	5	7	

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E
4	5

G	H
7	8
0 inv.	*0 inv.*

D	E	G	H
4	5	7	8

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

D	E
4	**5**

#invs = **1**

G	H
7	**8**

#invs = **1**

#inversions = **0**

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

#inversions in this part = **4**

D	E	G	H
4	5	7	8

#inversions in this part = **2**

#inversions from merging = **??**

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

#inversions in this part = **4**

D	E	G	H
4	**5**	**7**	**8**

#inversions in this part = **2**

#inversions from merging = **??**

MERGE AND COUNT

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

4 elements

D	E	G	H
4	**5**	**7**	**8**

$4 \geq 1$

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

3 elements

D	E	G	H
4	5	7	8

$4 \geq 2$

A							
1							

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E	G	H
4	5	7	8

2 elem.

$4 \geq 3$

A	B						
1	2						

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E	G	H
4	5	7	8

1 elem.

4 < 6

A	B	C					
1	2	3					

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

1 elem.

D	E	G	H
4	5	7	8

1 inv.

5 < 6

A	B	C	D				
1	2	3	4				

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

1 elem.

D	E	G	H
4	5	7	8
1 inv.	*1 inv.*		

7 ≥ 6

A	B	C	D	E			
1	2	3	4	5			

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E	G	H
4	5	7	8
1 inv.	*1 inv.*		

A	B	C	D	E	F		
1	2	3	4	5	6		

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E	G	H
4	5	7	8
1 inv.	*1 inv.*	*0 inv.*	

A	B	C	D	E	F	G	
1	2	3	4	5	6	7	

Counting Inversions

Divide and Conquer

A	B	C	F
1	2	3	6

D	E	G	H
4	5	7	8
1 inv.	*1 inv.*	*0 inv.*	*0 inv.*

A	B	C	D	E	F	G	H
1	2	3	4	5	6	7	8

Counting Inversions

Divide and Conquer

A	B	C	F
1	**2**	**3**	**6**

#inversions in this part = **4**

D	E	G	H
4	**5**	**7**	**8**

#inversions in this part = **2**

#inversions from merging = **2**

Counting Inversions

Divide and Conquer

A	B	C	D	E	F	G	H
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Total number of inversions = **8**