

Lecture 24

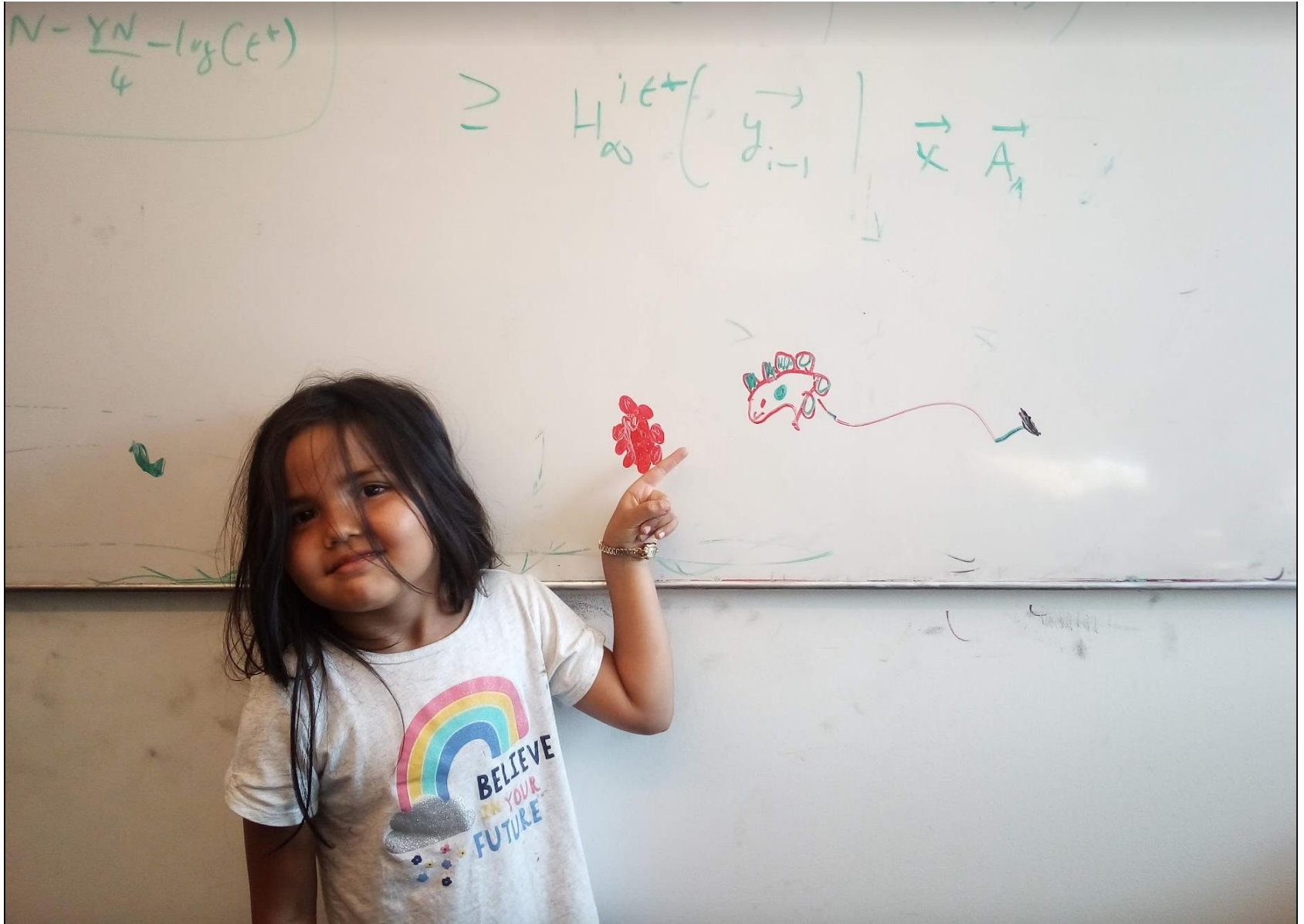
CSE 331

Oct 28, 2024

1st project deadline THIS Friday

Tue, Oct 15		(HW 4 out)
Wed, Oct 16	Dijkstra's algorithm     F24  F23  F22  F21  F19 x ²	[KT, Sec 4.4] Week 8 recitation notes
Fri, Oct 18	Correctness of Dijkstra's Algorithm  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 4.4] <i>Reading Assignment:</i> [KT, Sec 4.4]
Mon, Oct 21	Minimum Spanning Tree  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 4.5]
Tue, Oct 22		(HW 4 in, HW 5 out)
Wed, Oct 23	Cut Property Lemma  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 4.5] <i>Reading Assignment:</i> [KT, Sec 4.5, 4.6]
Fri, Oct 25	Mergesort  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 5.1]
Mon, Oct 28	Solving recurrence relations  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 5.1]
Tue, Oct 29		(HW 5 in)
Wed, Oct 30	Counting Inversions  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 5.3]
Fri, Nov 1	Multiplying large integers  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 5.5] (Project (Problems 1 & 2 Coding) in) <i>Reading Assignment:</i> Unraveling the mystery behind the identity
Mon, Nov 4	Closest Pair of Points  F23  F22  F21  F19  F18  F17 x ²	[KT, Sec 5.4] (Project (Problems 1 & 2 Reflection) in)

Questions/Comments?



Mergesort algorithm

Input: $a_1, a_2, \dots, a_n$

Output: Numbers in sorted order

MergeSort(a, n)

If $n = 1$ **return** the order a_1

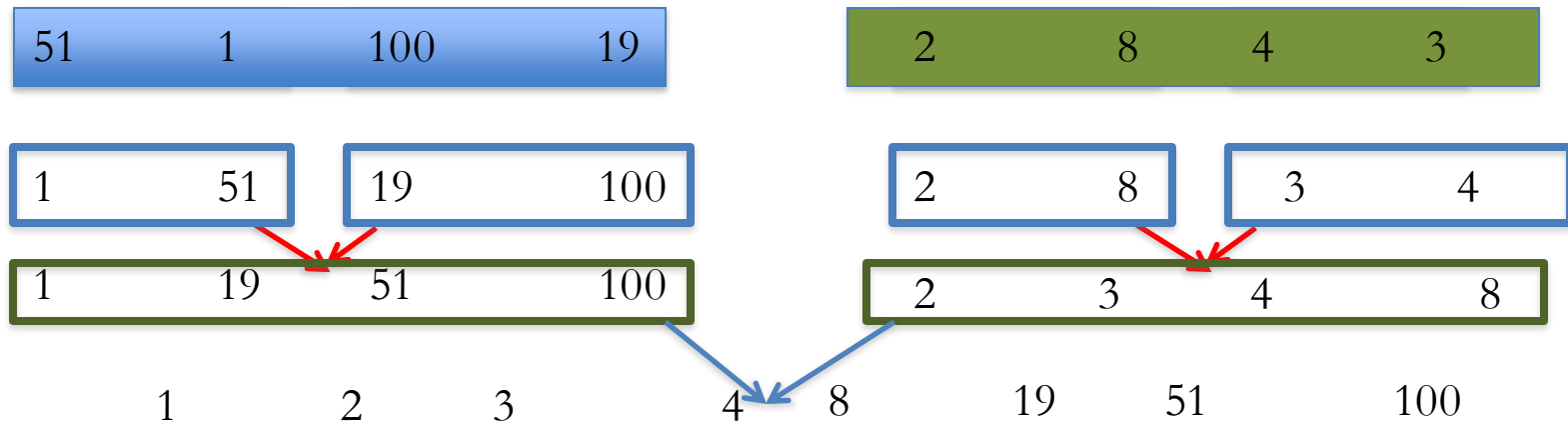
If $n = 2$ **return** the order $\min(a_1, a_2); \max(a_1, a_2)$

$a_L = a_1, \dots, a_{n/2}$

$a_R = a_{n/2+1}, \dots, a_n$

return MERGE (**MergeSort**($a_L, n/2$), **MergeSort**($a_R, n/2$))

An example run



MergeSort(a, n)

If $n = 1$ **return** the order a_1

If $n = 2$ **return** the order $\min(a_1, a_2); \max(a_1, a_2)$

$a_L = a_1, \dots, a_{n/2}$

$a_R = a_{n/2+1}, \dots, a_n$

return MERGE (MergeSort($a_L, n/2$), MergeSort($a_R, n/2$))

Correctness

Input: $a_1, a_2, \dots, a_n$

Output: Numbers in sorted order

MergeSort(a, n)

If $n = 1$ **return** the order a_1

If $n = 2$ **return** the order $\min(a_1, a_2); \max(a_1, a_2)$

$a_L = a_1, \dots, a_{n/2}$

$a_R = a_{n/2+1}, \dots, a_n$

return MERGE (MergeSort($a_L, n/2$), MergeSort($a_R, n/2$))

By
induction
on n

Inductive step follows from correctness of MERGE

Runtime analysis on the board...

