School of Computer Science

COMP SCI 1103/2103 Algorithm Design & Data Structure

Sorting

Sorting Algorithms

- Selection Sort
- Insertion Sort
- Bubble Sort
- Quicksort
- Merge Sort
- Bucket sort
- Heap sort

Selection Sort

- Select min value among unsorted part and swap to the corresponding position

- Input: list
  For i=1 to n-1 {
  j= index with min value among list[i] to list[n]
  swap list[i] and list[j]
  }

- Complexity
  - worst-case
  - average-case $O(n^2)$
  - best-case

Insertion Sort

- Insert value into the corresponding position among sorted part

- Input: list

```
For i=2 to n {
        For j =i-1 to 1 {
                t = list[i]
                If list[j] > t Then swap list[j] and list[j+1]
                Else break
        }
}
```

- Complexity
  - worst-case
  - average-case $O(n^2)$
  - best-case

Bubble Sort

- Larger values ‘bubble up‘ to the top of the list

- Input: list

```
For i=n to 2 {
    For j=1 to i-1 {
        If list[j] > list[j+1] Then swap list[j] and list[j+1]
    }
    If no swap Then break
}
```

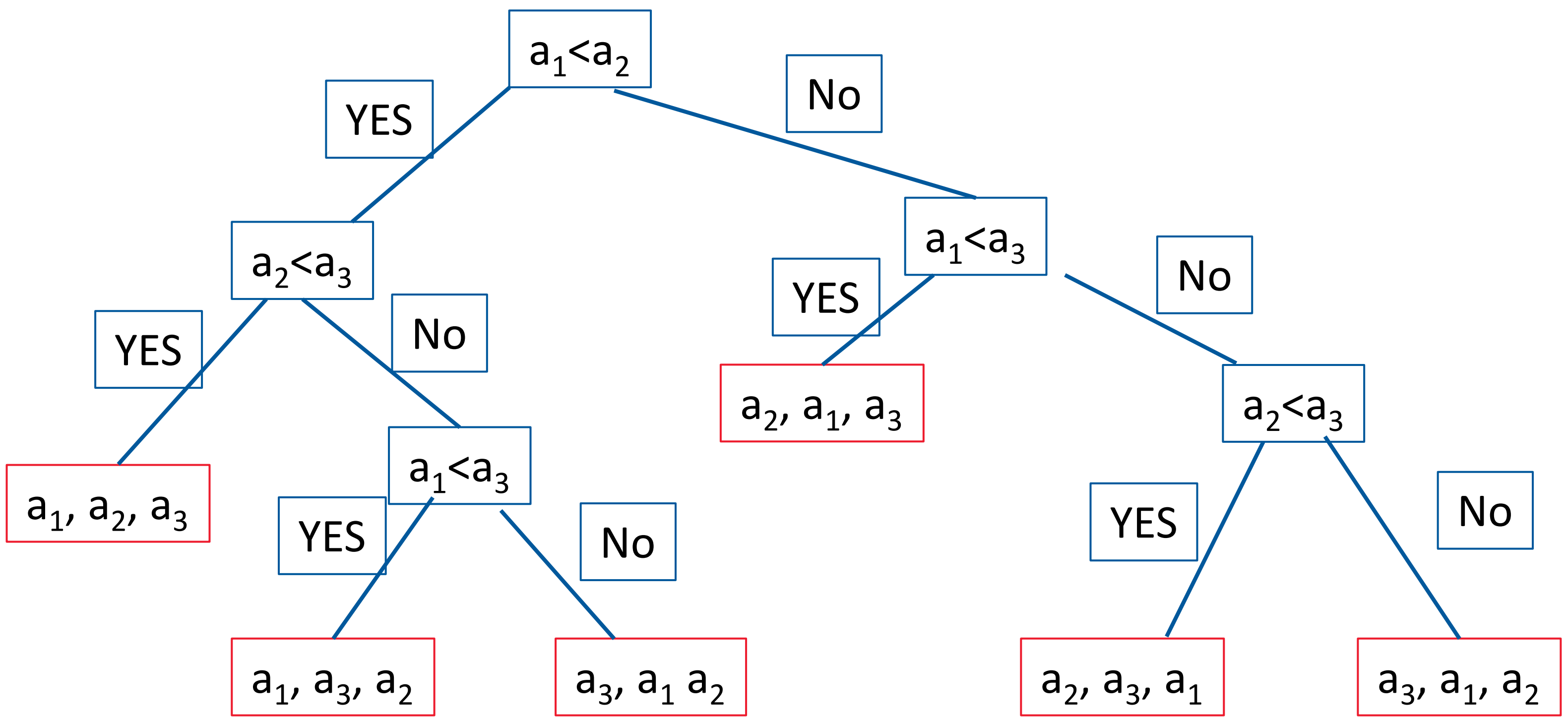
- Complexity
  - worst-case
  - average-case $O(n^2)$
  - best-case

Lower Bound Comparison-based Sorting Algorithms

- Assume that we have given n distinct elements $a_1, \ldots, a_n$.
- The algorithm must output a permutation of the elements $a_1, \ldots, a_n$.
- Example: Given input [2, 4, 1, 3] the output $[a_3, a_1, a_4, a_2]$ is the only correct one.
- For each input there is exactly one correct permutation if we have distinct elements.
- There are $n!$ permutations of the n elements that we can have as potential inputs.
- Let S be the set of the inputs consistent to the set of comparisons made so far. We have $|S|=n!$ at the beginning.

Comparison-based Algorithms

- Consider $[a_1, a_2, a_3]$



Lower Bound Comparison-based Sorting Algorithms

- A comparison splits S into two sets of inputs. One for which the answer would be YES and the other where the answer would be NO.
- Assume that an adversary always gives the answer for a comparison that results in the larger set S' after the split.
- Then we have to investigate a set S' with $|S'|>=|S|/2$.
- We must reduce our initial set S of size $|S|=n!$ to 1 and the number of comparisons for this is at least

$$\log_2(n!) = \log_2(n) + \log_2(n-1) + \ldots + \log_2(2) = \Omega(n \log n)$$

THE UNIVERSITY of ADELAIDE