Problem: function f: inputs $\longrightarrow$ outputs (or a relation if multiple outputs are correct)

an algorithm for more general cases must handle special cases too

SORT: given sequence of distinct integers, output order to select elts to put them in increasing order

SORT: $\{X \mid X \in \mathbb{Z}^* \text{ and } x_i = x_j \rightarrow i = j \text{ for all } i,j\} \rightarrow \mathbb{N}^*$
is defined by $SORT(X) = i_0, \ldots, i_{n-1}$ s.t. $i_0, \ldots, i_{n-1}$ is a permutation of $0, \ldots, n-1$
and $x_{i_0} < x_{i_1} < \cdots < x_{i_{n-1}}$

$x_1 = 2 \quad x_0 = 3 \quad x_4 = 5 \quad x_3 = 6 \quad x_2 = 8$

Ex: $SORT(3,2,8,6,5) = 1, 0, 4, 3, 2$
$SORT(703) = 0$

Lower Bound:

Problem: function f: inputs $\longrightarrow$ outputs (or a relation if multiple outputs are correct)

an algorithm for more general cases must handle special cases too

SORT : given sequence of distinct integers, output order to select elts to put them in increasing order

SORT : $\{X \mid X \in \mathbb{Z}^* \text{ and } x_i = x_j \rightarrow i=j \text{ for all } i,j\} \rightarrow \mathbb{N}^*$
is defined by SORT$(X) = i_0, \dots, i_{n-1}$ s.t. $i_0, \dots, i_{n-1}$ is a permutation of $0, \dots, n-1$ and $x_{i_0} < x_{i_1} < \dots < x_{i_{n-1}}$

Algorithm A solves problem X if $A(i) = X(i)$ for all inputs i

Solutions to SORT: mergesort, quicksort, bubble sort, insertion sort, selection sort, shell sort, heap sort, tree sort, Timsort, comb sort, monkey sort, ...

$\Theta(n \log n)$ $\Theta(n^2)$

worst case always $\Omega(n \log n)$

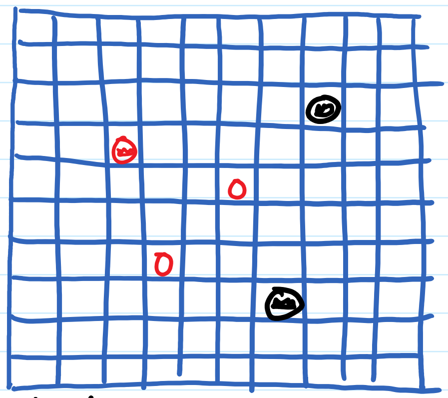
Lower Bound: $f(n)$ such that any solution for X has worst case $\Omega(f(n))$

Lower bound for SORT is $n \log n$

Problems with Polynomial Solutions

Polynomial: sort, maximum, stable matching, counting inversions, interval selection, shortest paths, MST, bipartite matching, maximum flow

Not Polynomial:



Black to move

Generalized Checkers: given position, who has winning strategy?

worst case $\Omega(c^n)$ for some c (requires exponential time)

Undecidable: HALT – given program P and input x, does P go into infinite loop on x?

↳ no algorithm to solve

Lower Bound (asymptotic): $f(n)$ such that any solution for X has worst case $\Omega(f(n))$

SORT: lower bound $n \log n$

MAXIMUM: lower bound $n-1$ (exact, and lower bound on best case too)

algorithms that handle only special cases can beat these

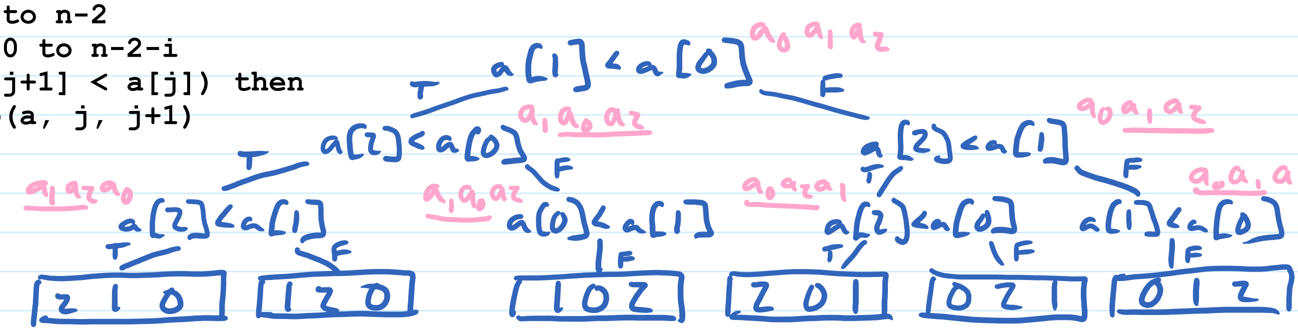
counting sort: $\Theta(n)$ if the inputs are in $0, \ldots, n-1$

bucket sort: expected $\Theta(n)$ if inputs are uniformly randomly distributed over $[0,1)$

Decision Trees

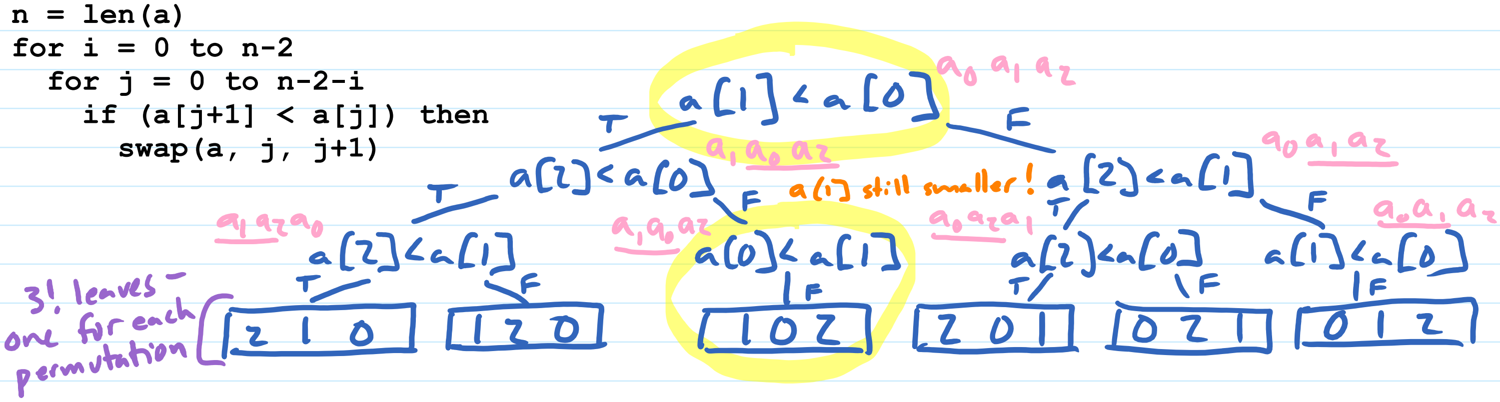
Decision tree: tree of all execution paths with nodes representing branches based on evaluation of inputs

```
BUBBLESORT(a)
n = len(a)
for i = 0 to n-2
  for j = 0 to n-2-i
    if (a[j+1] < a[j]) then
      swap(a, j, j+1)
```



Decision tree: tree of all execution paths with nodes representing branches based on evaluation of inputs

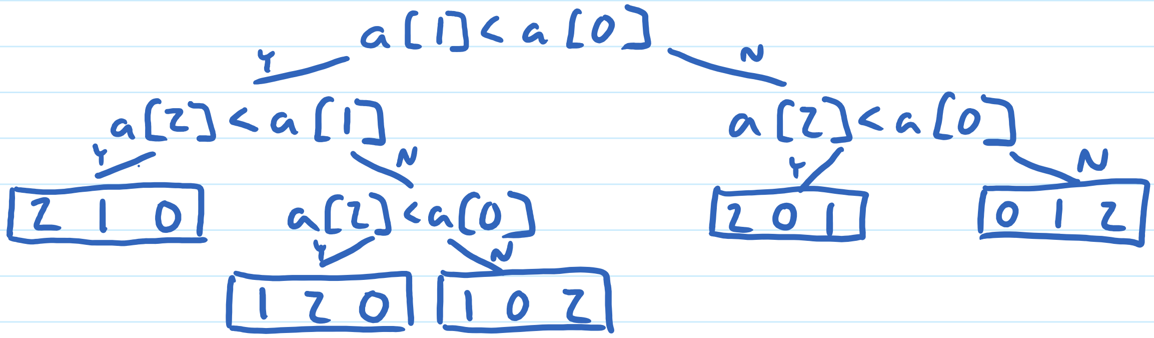
```
BUBBLESORT(a)
n = len(a)
for i = 0 to n-2
  for j = 0 to n-2-i
    if (a[j+1] < a[j]) then
      swap(a, j, j+1)
```



$n!$ permutations $\rightarrow \geq n!$ leaves $\rightarrow$ height $\geq \log_2 n! \rightarrow$ height $\in \Omega(n \log n) \rightarrow$ worst case $\Omega(n \log n)$

information theoretic lower bound

Missing Leaf



For which input does this decision tree give the incorrect output?

a) 30, 20, 10

b) 5, 10, 2

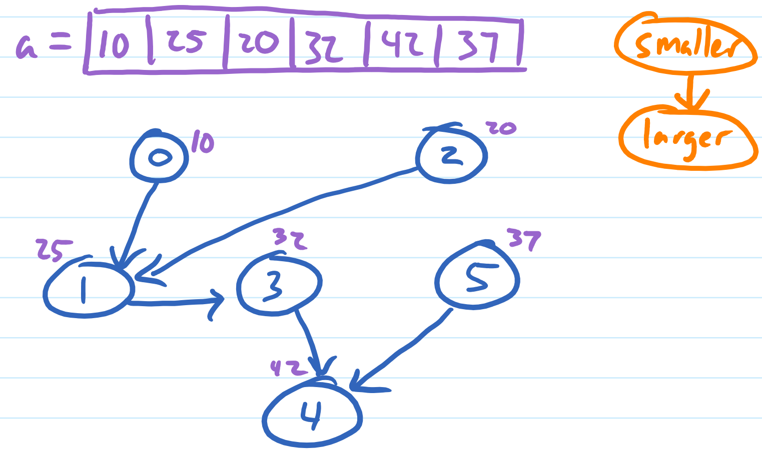
c) 10, 20, 30

d) 4, 8, 5

Comparison Graphs

Comparison Graph: records results of comparisons of inputs

```
MAXIMUM(a)
n = len(a)
max = 0
for i = 1 to n-1
   if (a[i] > a[max]) then
      max = i
return max
```

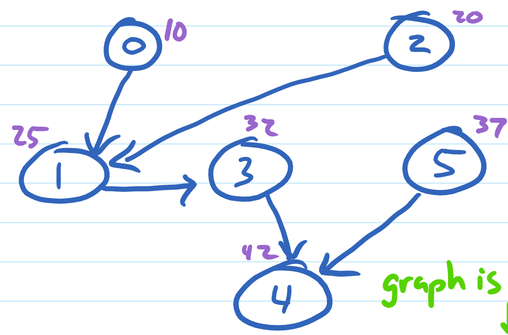


Comparison Graph: records results of comparisons of inputs

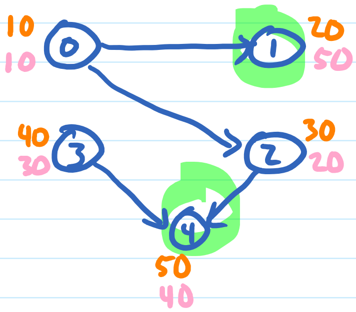
```
MAXIMUM(a)
n = len(a)
max = 0
for i = 1 to n-1
   if (a[i] > a[max]) then
      max = i
return max
```

a = | 10 | 25 | 20 | 32 | 42 | 37 |

smaller
↓
larger



graph is connected
↓
has n-1 edges
↓
worst case is n-1 comparisons

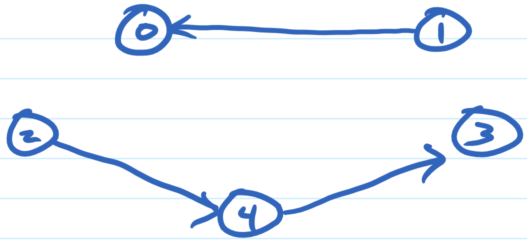


input 1
input 2

output : a[4] is max
wrong for input 2 !

Missing Comparison

A purported solution for MAXIMUM terminated with the following comparison graph and output.



output: a[3] is the max

Which of the following inputs must it give the wrong answer for?

a) 3, 2, 4, 8, 6

b) 1, 2, 3, 4, 5

c) 15, 11, 4, 8, 6

d) 5, 1, 4, 8, 6