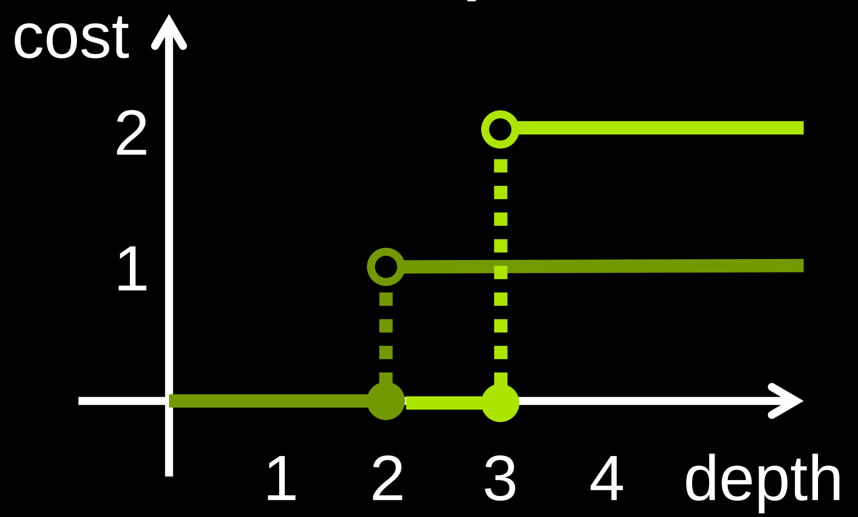
An Algorithm for the Huffman Tree Problem with Unit Step Functions



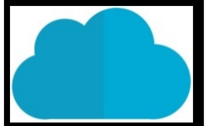
Hiroshi Fujiwara (Shinshu University)
Takuya Nakamura (Toyohashi University of Technology)
Toshihiro Fujito (Toyohashi University of Technology)

This talk is based on:

- Hiroshi Fujiwara, Takuya Nakamura, and Toshihiro Fujito. **The Huffman Tree Problem with Unit Step Functions.** IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E98-A, No. 6, IEICE, June, 2015. To appear.
- Previous Work: Hiroshi Fujiwara and Tobias Jacobs. **On the Huffman and Alphabetic Tree Problem with General Cost Functions.** Algorithmica, Vol. 69, Issue 3, pp. 582--604, Springer, July, 2014.
  - Conference version in ESA2010.

preprint available at: http://goo.gl/a5GWFG

Huffman Code [Huffman1952]

event					example of cordword sequence	average cordword length
probability	$\frac{1}{10}$	$\frac{3}{10}$	$\frac{5}{10}$	$\frac{1}{10}$		
simple code	00	01	10	11	101001100010 100100101110	2.0
Huffman code	000	01	1	001	110110001101 00010011	1.7

Huffman's Algorithm [Huffman1952]

sorted by probability →

$O(n \log n)$ time

$1/10_{(1)}$ $1/10_{(4)}$ $2/10_{(1,4)}$ $3/10_{(2)}$ $5/10_{(1,2,4)}$ $5/10_{(3)}$




000 001 (00) 01 (0) 1

Huffman Tree Problem

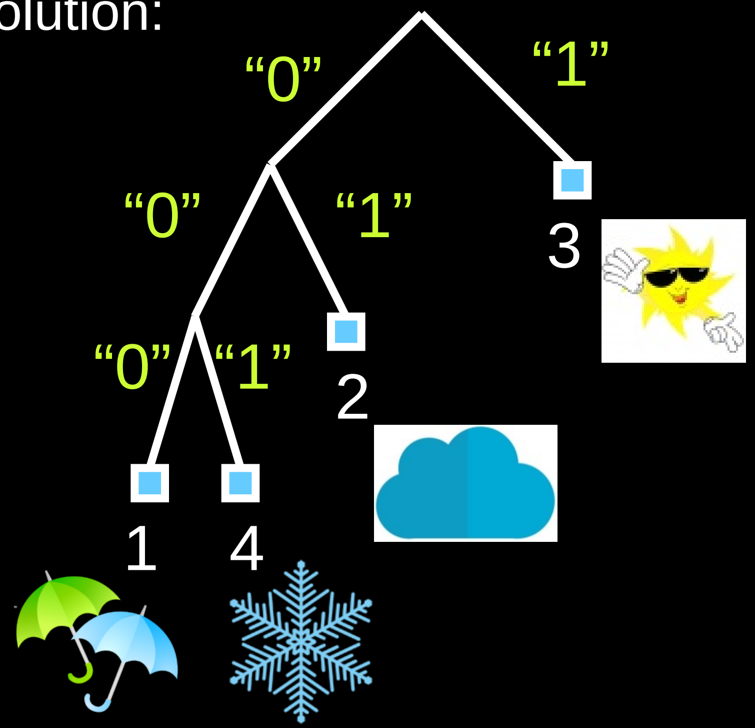
Huffman Tree Problem

Input: weights $w_1, w_2, ..., w_n$
Output: binary tree
Objective:

$$\sum_{i=1}^{n} w_i d_i \rightarrow \text{minimize}$$

Example: $w_1 = 1, w_2 = 3, w_3 = 5, w_4 = 1$

Solution:



Huffman code:
1: "000"
2: "01"
3: "1"
4: "001"

Huffman Tree Problem

Huffman Tree Problem

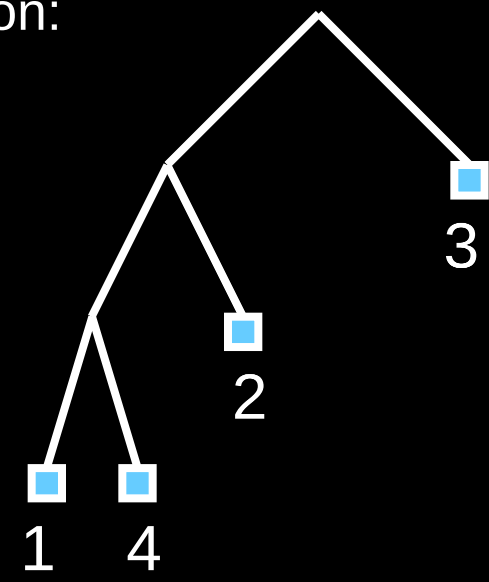
Input: weights $w_1, w_2, ..., w_n$
Output: binary tree
Objective:

$$\sum_{i=1}^{n} w_i d_i \quad \rightarrow \text{minimize}$$

d_i = depth of leaf i

Example: $w_1 = 1$, $w_2 = 3$, $w_3 = 5$, $w_4 = 1$

Solution:



objective value = 3 + 3 + 6 + 5 = 17

Alphabetic Tree Problem [Gilbert&Moore1959]

Alphabetic Tree Problem

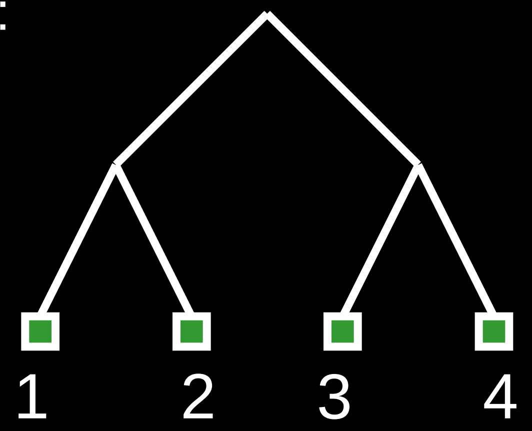
Input: weights $w_1, w_2, \ldots, w_n$
Output: binary tree (leaf order fixed)
Objective:

$$\sum_{i=1}^{n} w_i d_i \quad \rightarrow \text{minimize}$$

d_i = depth of leaf i

Example: $w_1 = 1$, $w_2 = 3$, $w_3 = 5$, $w_4 = 1$

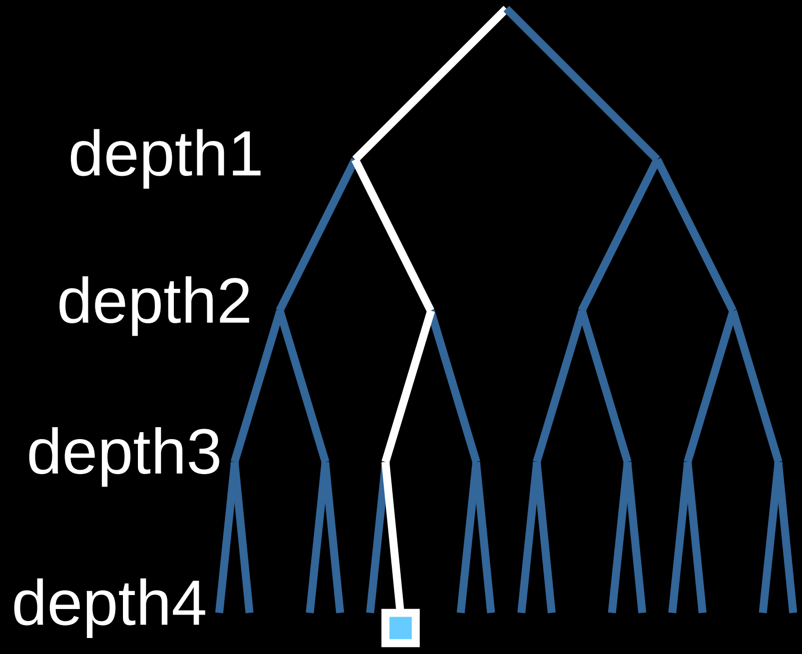
Solution:



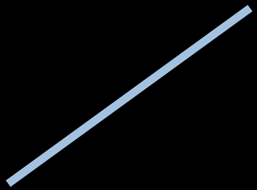
leaf order should be $w_1, w_2, \ldots, w_n$

objective value = 2 + 6 + 10 + 2 = 20

Generalize Objective Function



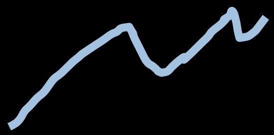
Objective: $\sum_{i=1}^{n} w_i d_i$



contribution of a leaf = $w_i d_i$

generalize

Objective: $\sum_{i=1}^{n} f_i(d_i)$



contribution of a leaf = $f_i(d_i)$

Huffman Tree Problem

Input: weights $w_1, w_2, \ldots, w_n$
Output: binary tree (leaf order arbitrary)
Objective:

$$\sum_{i=1}^{n} w_i d_i \rightarrow \text{minimize}$$

generalize →

General Cost Huffman Tree Problem

Input: functions $f_1, f_2, \ldots, f_n$
Output: binary tree (leaf order arbitrary)
Objective:

$$\sum_{i=1}^{n} f_i(d_i) \rightarrow \text{minimize}$$

Alphabetic Tree Problem

Input: weights $w_1, w_2, \ldots, w_n$
Output: binary tree (leaf order fixed)
Objective:

$$\sum_{i=1}^{n} w_i d_i \rightarrow \text{minimize}$$

generalize →

General Cost Alphabetic Tree Problem

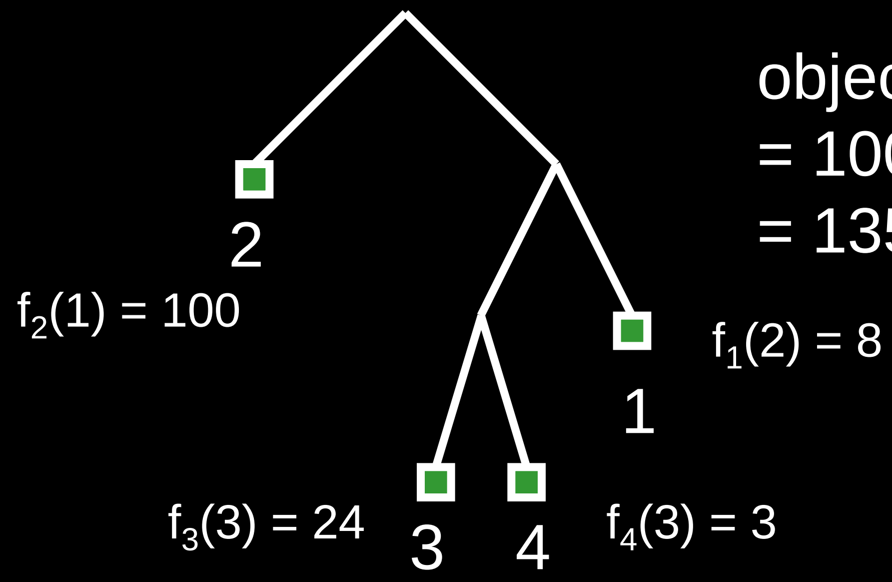
Input: functions $f_1, f_2, \ldots, f_n$
Output: binary tree (leaf order fixed)
Objective:

$$\sum_{i=1}^{n} f_i(d_i) \rightarrow \text{minimize}$$

Instance of General Cost Huffman Tree Problem

Example: $f_1(x) = x^3$, $f_2(x) = 100x$, $f_3(x) = 8x$, $f_4(x) = x$

Solution:



objective value
= 100 + 24 + 3 + 8
= 135

Previous Results on General Cost Huffman Tree Problem

leaf order
arbitrary

class of cost functions	time complexity
linear	$O(n \log n)$ [Huffman1952]
non-decreasing convex	$O(n^2 \log n)$ [Fujiwara&Jacobs2014]
general	NP-hard (even for {0,1}-functions) [Fujiwara&Jacobs2014]

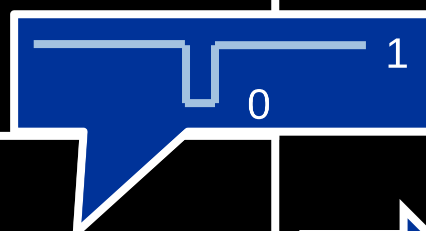
Results on General Cost Alphabetic Tree Problem

class of cost functions		time complexity
subtree optimality & monotonicity	linear	$O(n \log n)$ [Hu&Kleitman&Tamaki1979]
	exponential	$O(n \log n)$ [Hu&Kleitman&Tamaki1979]
	general	$O(n^2)$ [Fujiwara&Jacobs2014]
subtree optimality		$O(n^3)$ [Fujiwara&Jacobs2014]
monotonicity	non-decreasing convex	$O(n^2 \log n)$ [Fujiwara&Jacobs2014]
	general	$O(n^3)$ [Fujiwara&Jacobs2014]
general		$O(n^4)$ [Fujiwara&Jacobs2014]

leaf order fixed

Motivation of Our New Work

class of cost functions	time complexity
linear	$O(n \log n)$
non-decreasing convex	$O(n^2 \log n)$
general	NP-hard (even for {0,1}-functions)

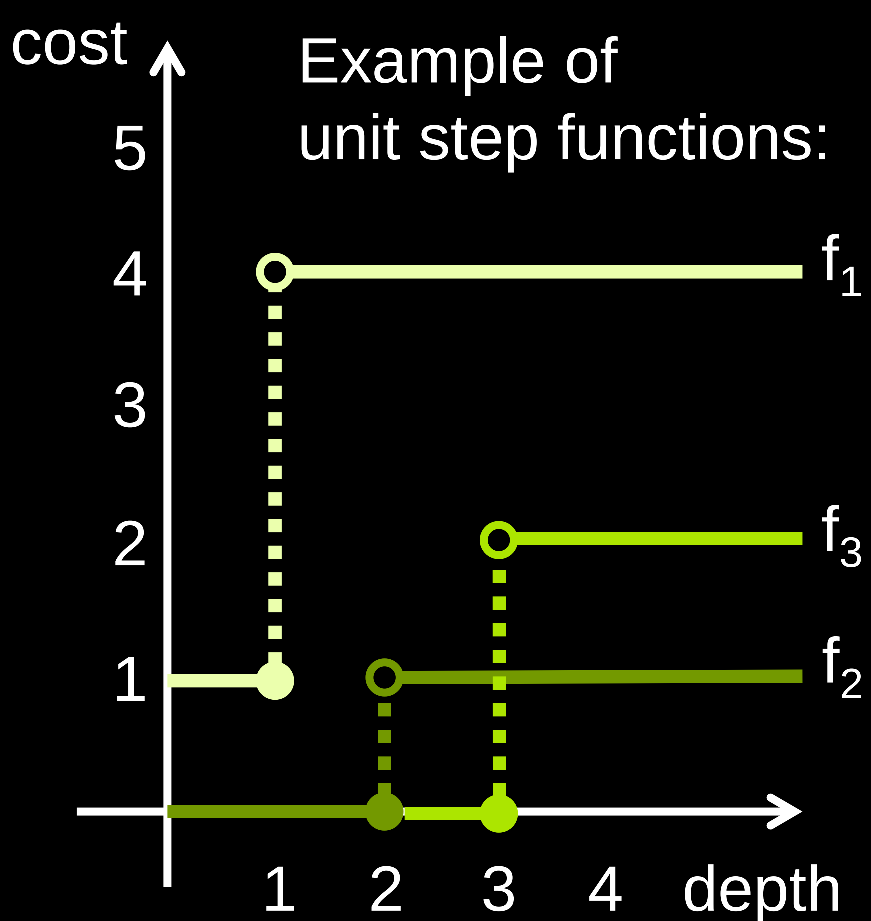


unit step functions solvable in $O(n \log n)$ time!

non-decreasing {0,1}-functions solvable in $O(n \log n)$ time!

any class of {0,1}-functions solvable in polynomial time?

Motivation of Our New Work



unit step functions solvable in $O(n \log n)$ time!

↑

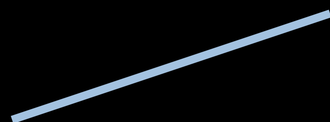
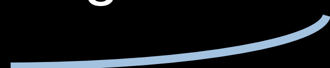
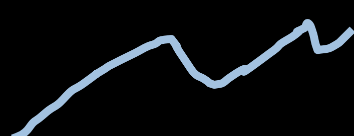
non-decreasing {0,1}-functions solvable in $O(n \log n)$ time!

↑

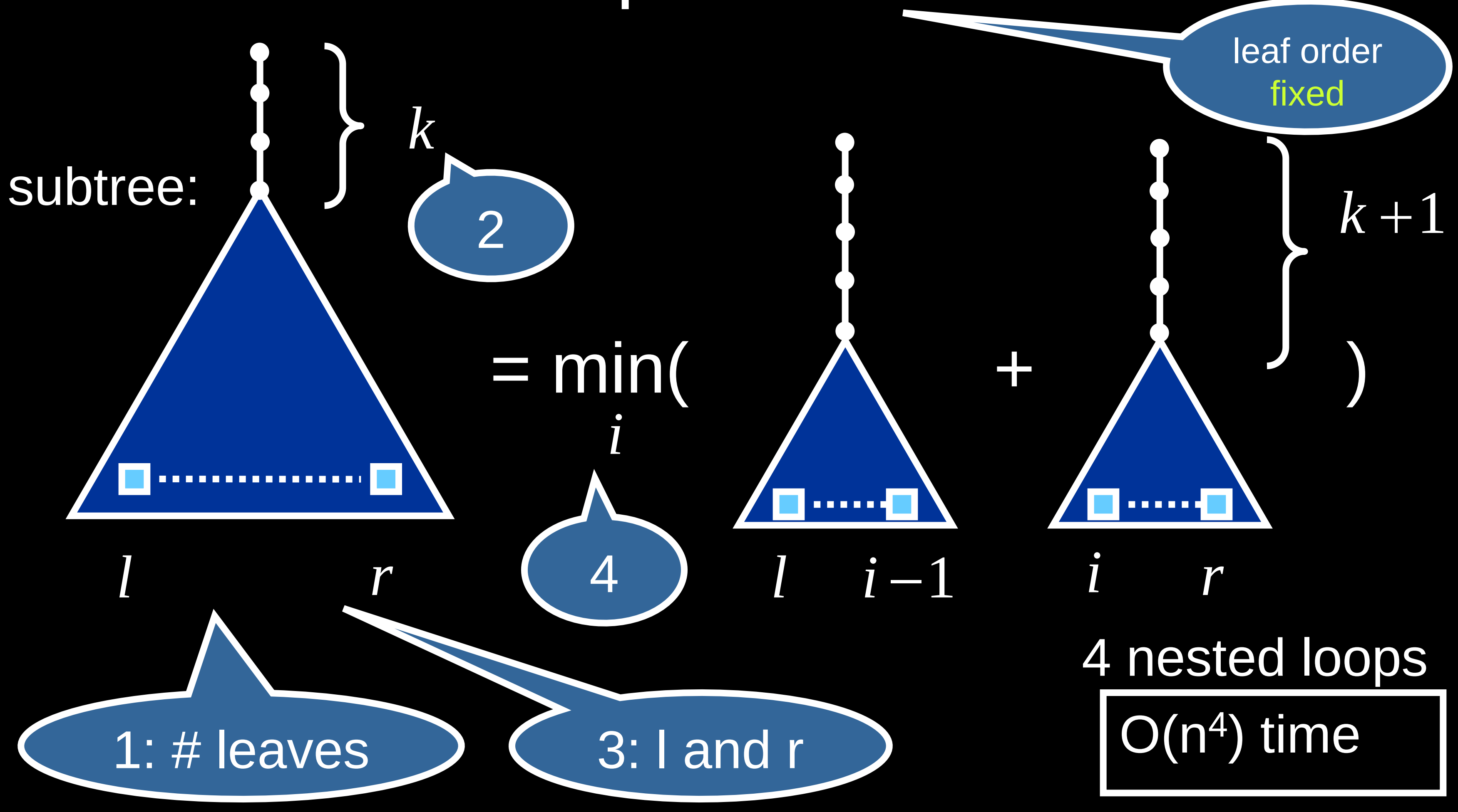
any class of {0,1}-functions solvable in polynomial time?

Our Result on General Cost Huffman Tree Problem

leaf order arbitrary

class of cost functions	time complexity
linear 	$O(n \log n)$ [Huffman1952]
unit step 	$O(n \log n)$ [Fujiwara&Nakamura&Fujito2015]
non-decreasing convex 	$O(n^2 \log n)$ [Fujiwara&Jacobs2014]
general 	NP-hard (even for {0,1}-functions) [Fujiwara&Jacobs2014]

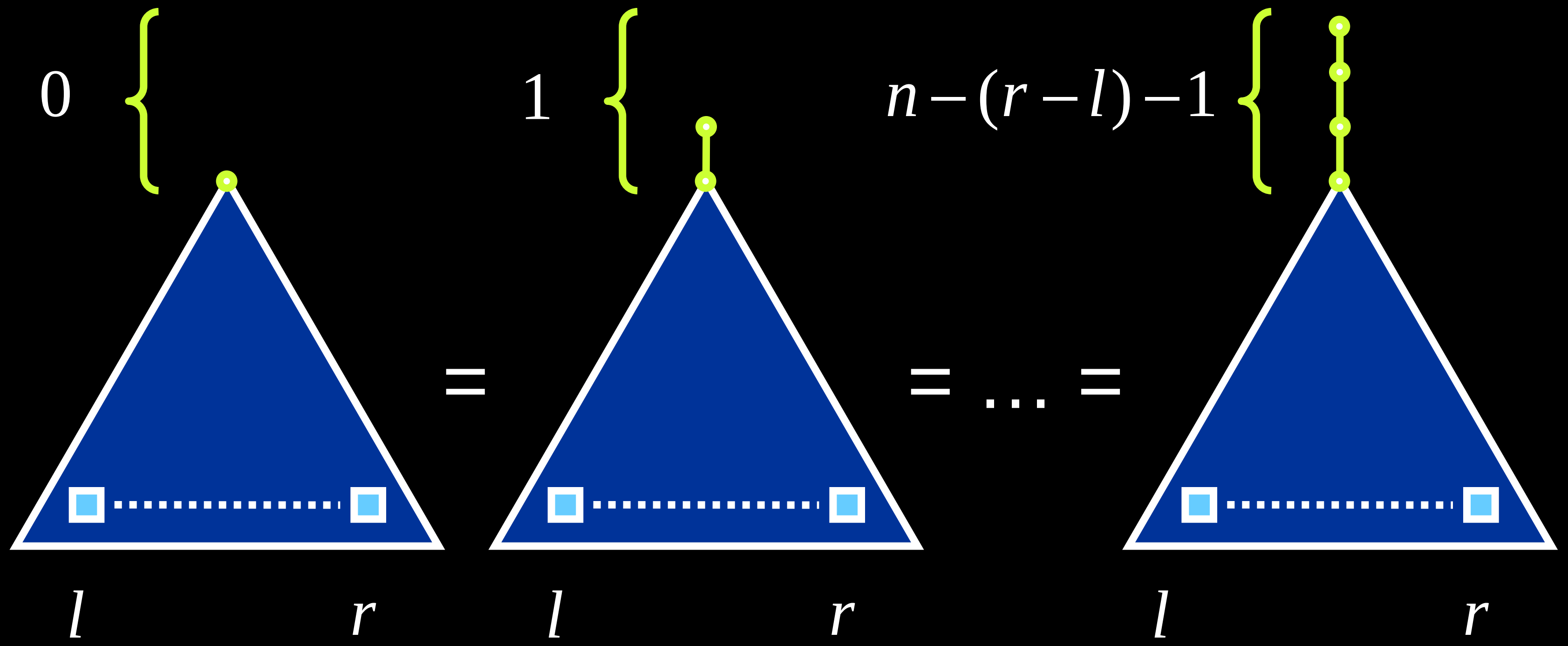
Dynamic Programming for General Cost Alphabetic Tree Problem



4 nested loops

O(n^4) time

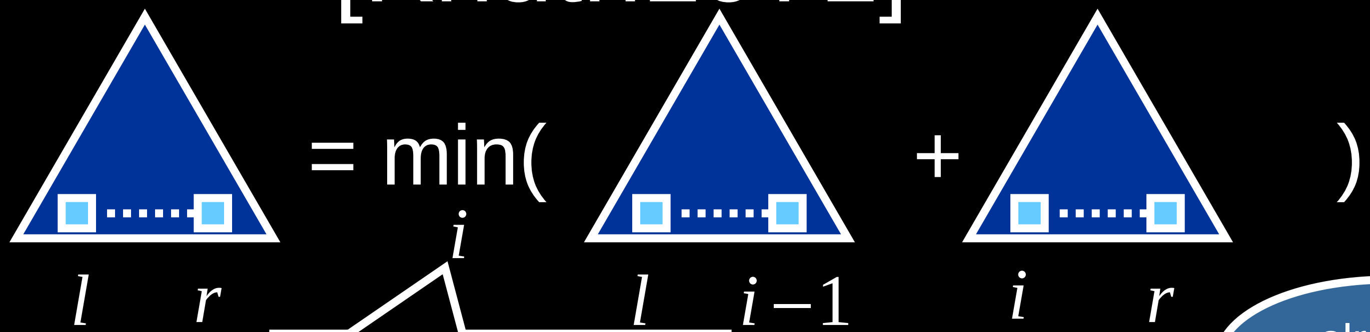
Speed Up: Subtree Optimality



Same subtree is optimal regardless of path to root

$O(n^3)$ time

Speed Up: Monotonicity [Knuth1971]



i[l,r,k] := minimizer

already calculated

i[l,r-1,k] ≦ i[l,r,k] ≦ i[l+1,r,k] holds

already calculated

iteration for subproblems (1,a,k),(2,a+1,k),…,(n-a+1,n,k) =

$$\sum_{l=1}^{n-a} (i[l+1, l+a, k] - i[l, l+a-1, k] + 1)$$

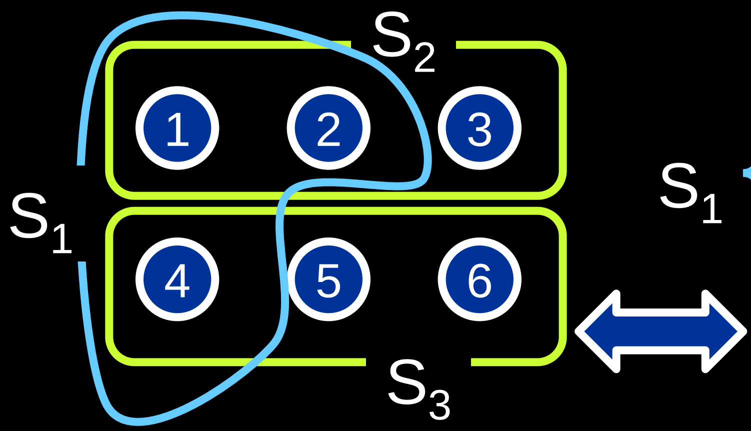
$$= i[n-a+1, n, k] - i[1, a, k] + n - a$$

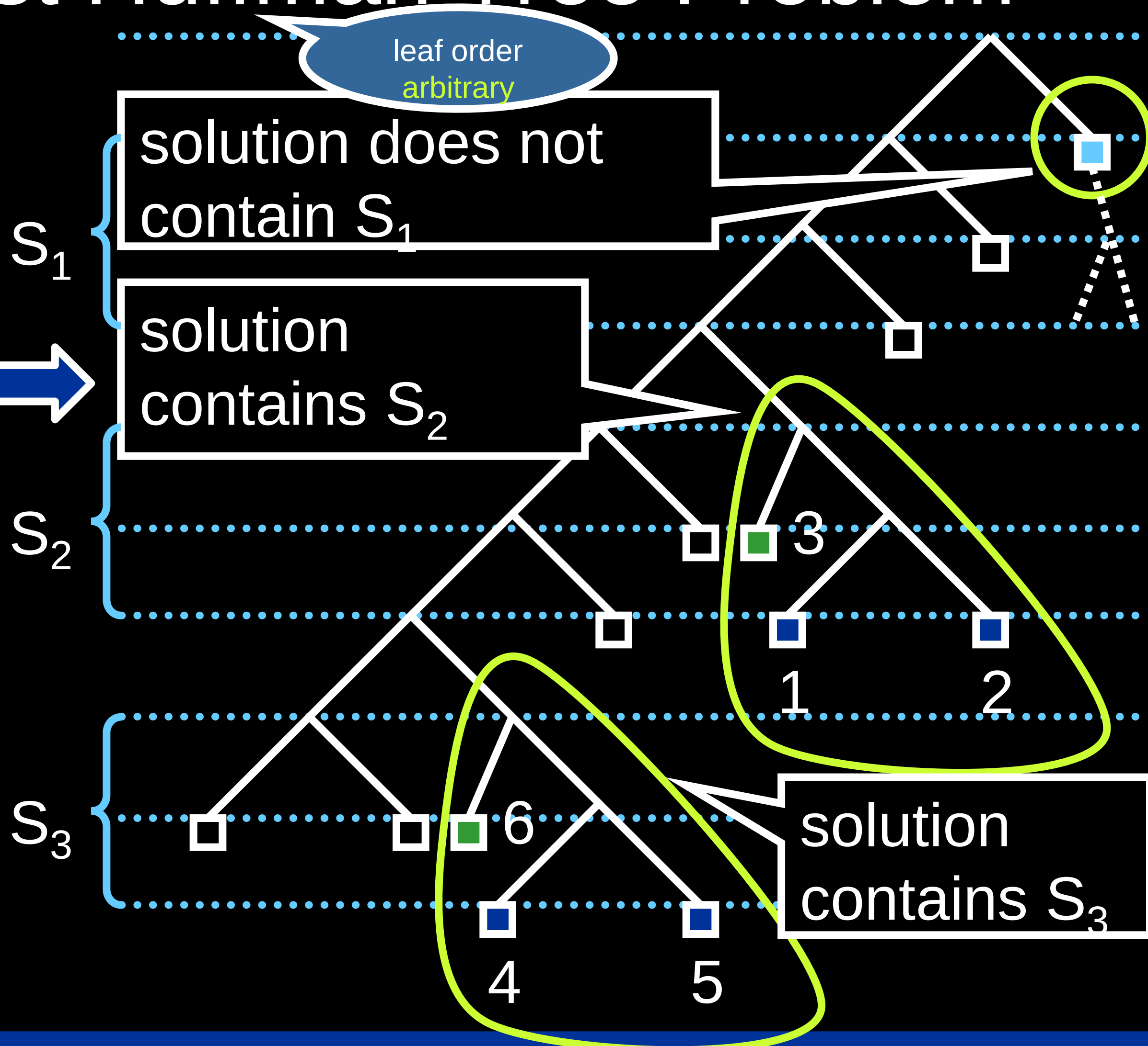
$$\leq 2n$$

amortizedly

O(n^3) time

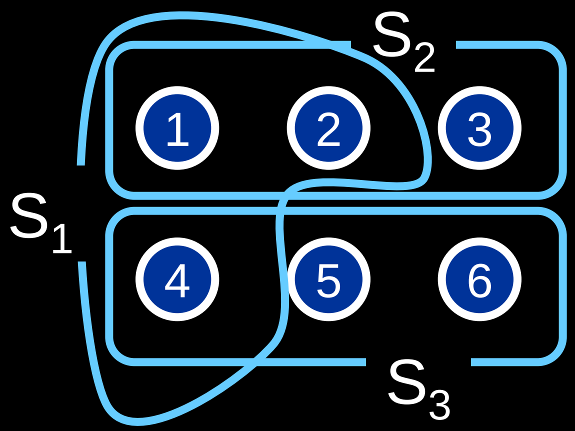
Exact Cover by 3-Sets is reduced to General Cost Huffman Tree Problem





Basic Idea:
Use functions such that:
f = 0 for some depths,
f = 1 otherwise

Exact Cover by 3-Sets is reduced to General Cost Huffman Tree Problem



NP-complete
[Karp1972]

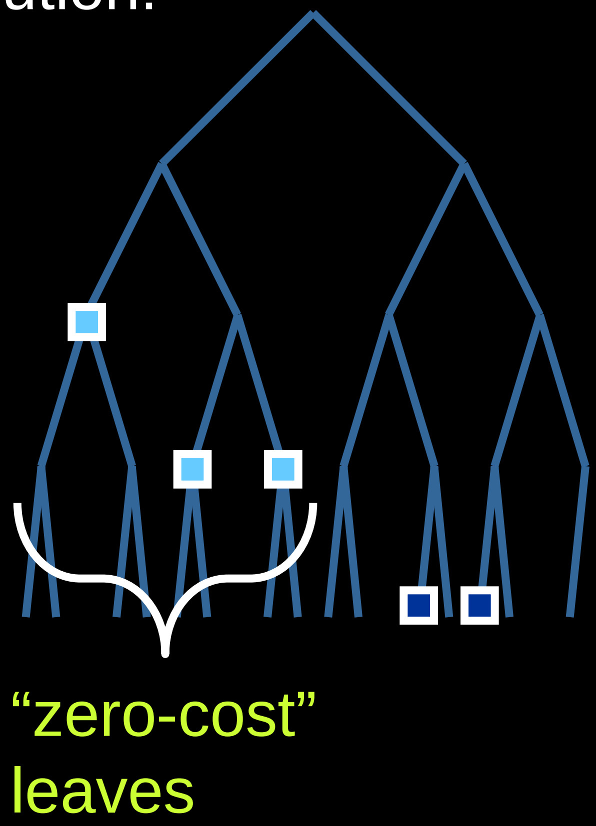
YES-instance of Exact Cover by 3-Sets
⇔
Instance of General Cost Huffman Tree Problem admits objective value = 0

Theorem

General Cost Huffman Tree Problem is NP-hard

Key Idea for Unit Step Functions

Solution:



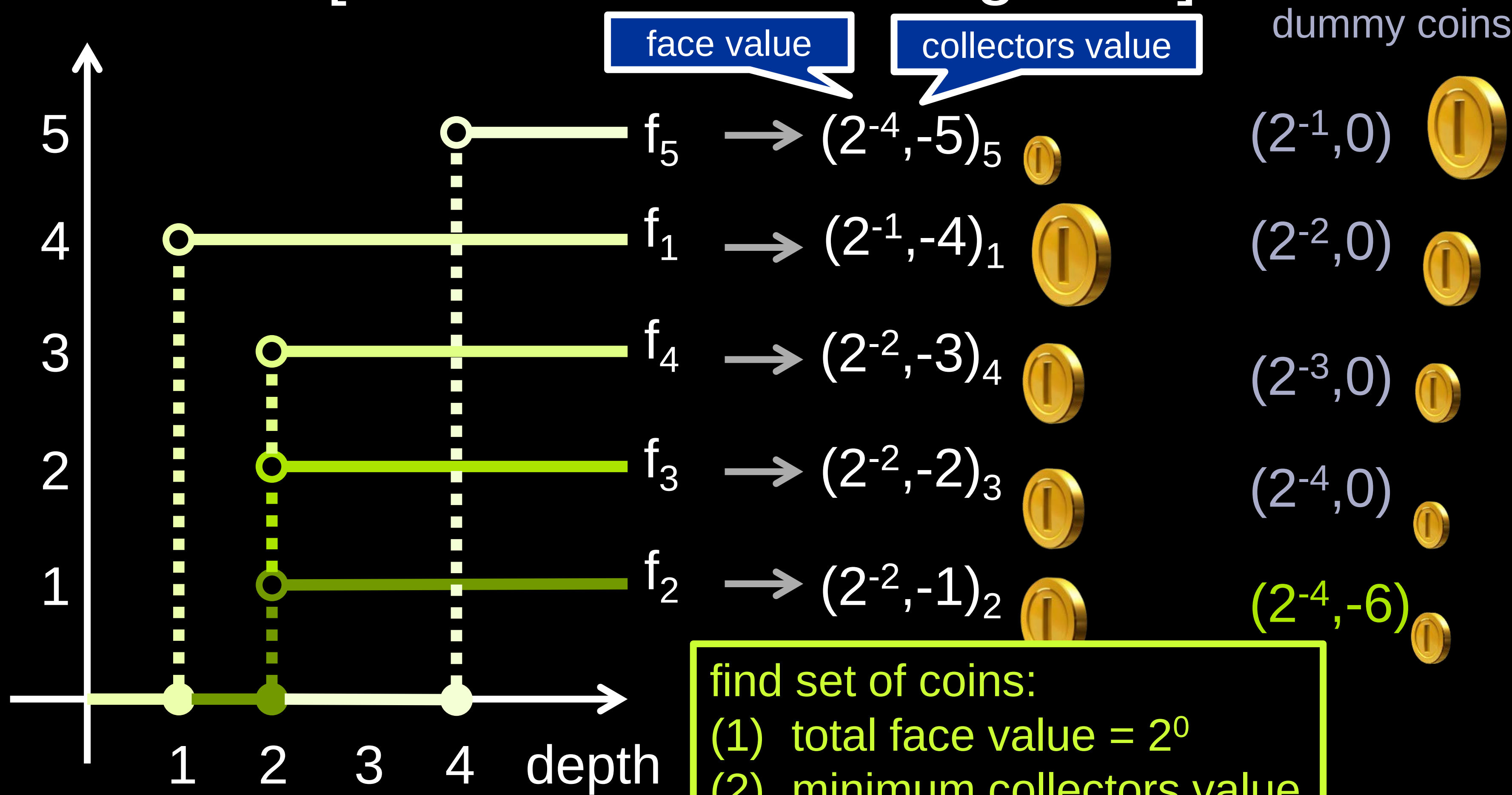
- Find "zero-cost" leaves (= no contribution to objective function)
- Formulate Coin Collector's Problem [Larmore&Hirschberg1990]
  - Find set of coins:

(1) total face value fixed

(2) minimum collectors value

(special case of KNAPSACK)

Reduction to Coin Collector's Problem [Larmore&Hirschberg1990]

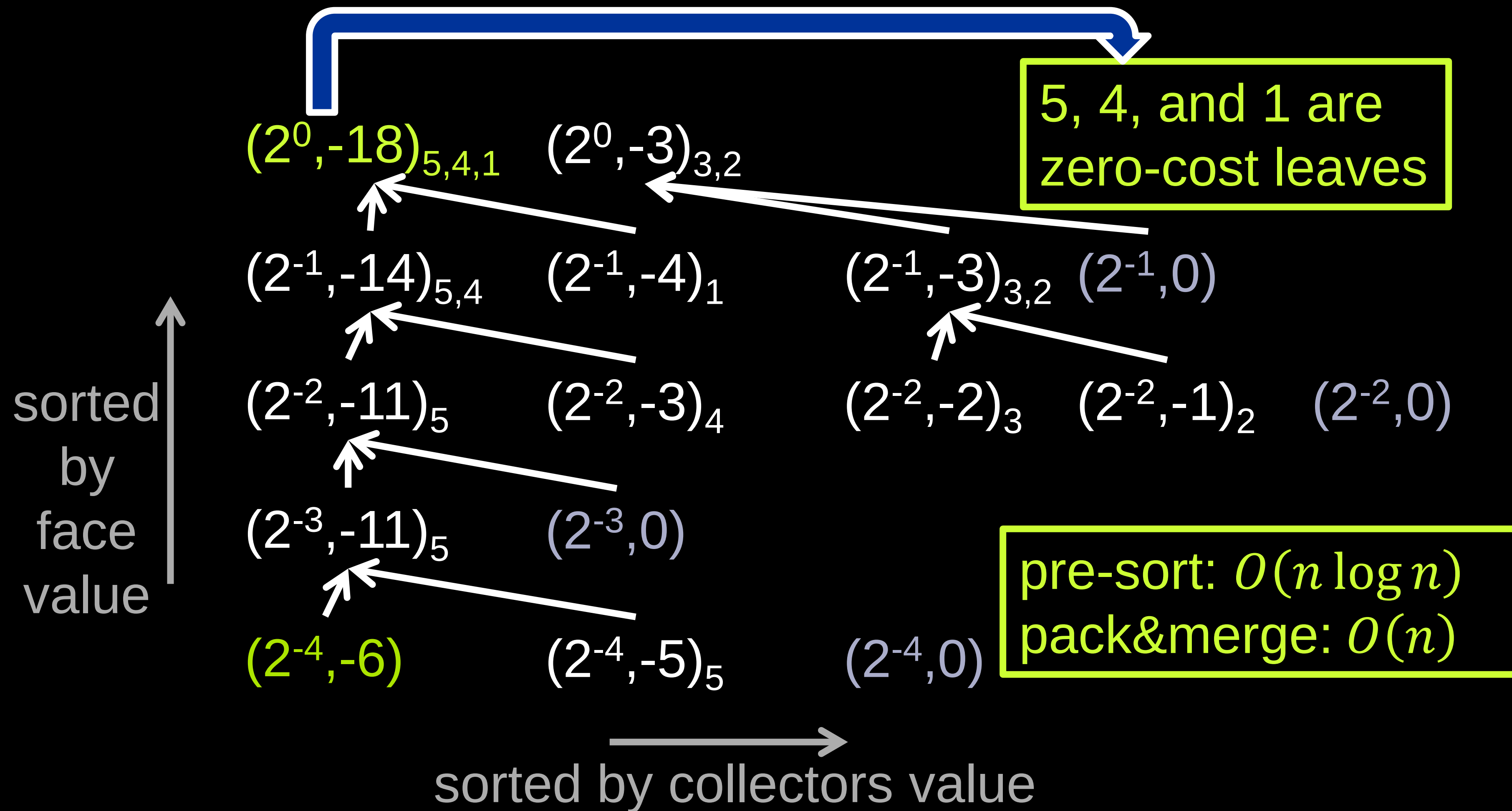


find set of coins:

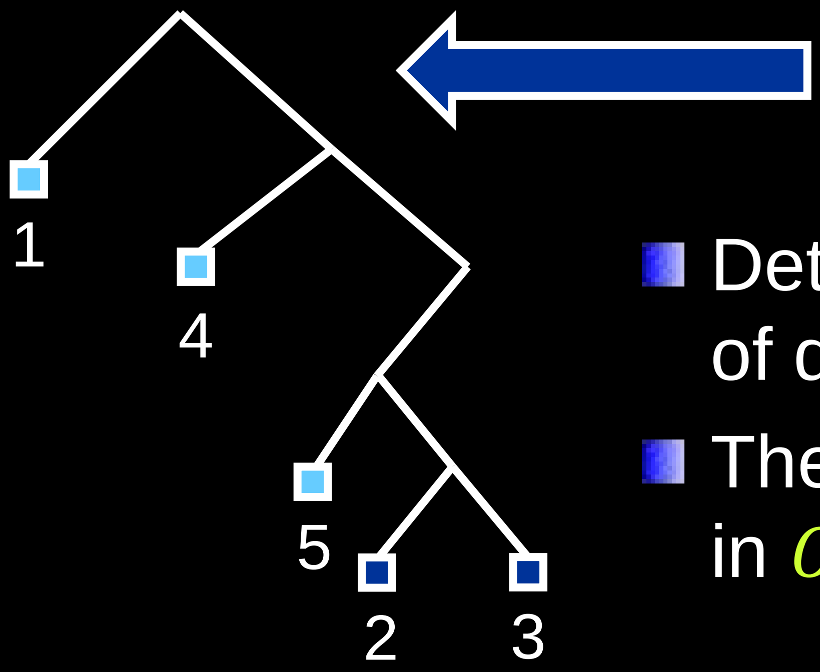
(1) total face value = 2^0

(2) minimum collectors value

PackageMerge Algorithm [Larmore&Hirschberg1990]



Solution for Unit Step Functions

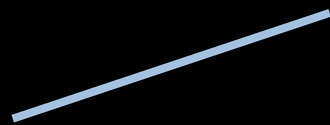
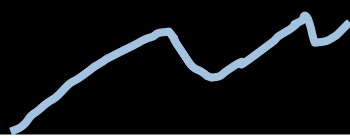


5, 4, and 1 are zero-cost leaves

- Determine sequence of depths
- Then, construct tree in $O(n)$ time

objective value = 1 + 2 = 3

Summary

class of cost functions	time complexity
linear 	$O(n \log n)$ [Huffman1952]
unit step 	$O(n \log n)$ [Fujiwara&Nakamura&Fujito2015]
general 	NP-hard [Fujiwara&Jacobs2014]

Future Works

- What functions are solvable in $O(n \log n)$ time?
- Full binary tree for decreasing {0,1}-functions?