Fundamental Algorithmic Techniques V

October 18, 2025

Outline

The greedy algorithm paradigm

Characteristics of greedy algorithms

The greedy algorithm paradigm

Best possible (greedy) choice right now, for immediate best outcome!

Requirements:

1. **greedy-choice property:**
   globally optimal solution $\Leftrightarrow$ local optimal (greedy) choices
2. **optimal substructure**

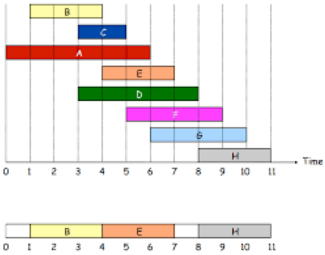
Examples where Greedy Algorithm is suboptimal

- life!?
- road...
- 0-1 knapsack problem

Examples with Greedy: Courses Allocation

Course allocation:
For starting time T:

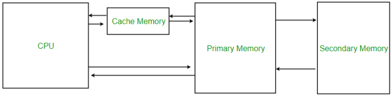
- Select out courses with starting $< \mathrm{T}$
- Choose remaining course C with lowest start time T_{end}
- Update $\mathrm{T} \leftarrow C_{T_{\text{end}}}$

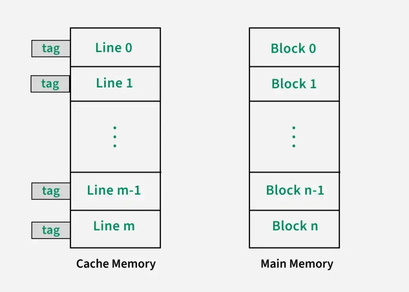


Examples with Greedy: Cache Memory Management

On request for block b_i:

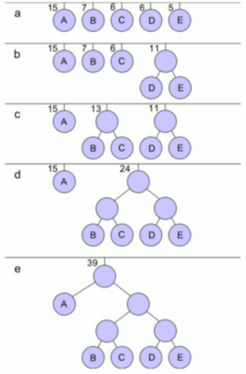
- **Hit:** b_i is in cache $\to$ no change.
- **Miss, cache not full:** add b_i.
- **Miss, cache full:** evicts one block, add b_i.

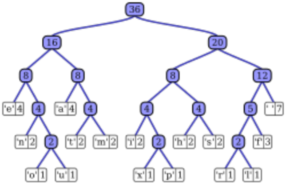




Greedy Strategy for cache allocation removing less used cache blocks

Huffman Encoding Example





Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

Huffman Code: Numerical Example

Input (A, W)	Symbol (a_i)					
	a	b	c	d	e	**Sum**
Weights (w_i)	0.10	0.15	0.30	0.16	0.29	$= 1$
Output C	**Codewords** (c_i)					
	010	011	11	00	10	
Codeword length (ℓ_i)	3	3	2	2	2	
$\ell_i w_i$	0.30	0.45	0.60	0.32	0.58	$L(C) = 2.25$
Optimality	**Probability budget** $(2^{-\ell_i})$					
	1/8	1/8	1/4	1/4	1/4	$= 1.00$
Info. content $(-\log_2 w_i)$	3.32	2.74	1.74	2.64	1.79	
$-w_i \log_2 w_i$	0.332	0.411	0.521	0.423	0.518	$H(A) = 2.205$

Huffman coding approximates the optimal lossless compression bound!

- The Huffman code minimizes the expected length: $L(C) = \sum_i w_i \, \ell_i$
- The (Shannon) entropy of the source is: $H(A) = -\sum_i w_i \log_2 w_i$
- Huffman coding is near-optimal: $H(A) \leq L(C) < H(A) + 1$

Characteristics of greedy algorithms

Greedy stays ahead!

- $A = \{a_0, a_1, \ldots, a_n\}$ algo. sequence and $O = \{o_0, o_1, \ldots, o_n\}$ optimal sequence
- choose a measure $\mu(\dot{})$
- show stay ahead:
  $\mu(a_0, a_1, \ldots, a_n) \leq \mu(o_0, o_1, \ldots, o_n)$ by induction!
- Prove optimality: use greedy stays ahead to generate contradiction