Greedy Algorithms

"Greedy algorithms build up a solution piece by piece always choosing the next piece that offers the most obvious and immediate benefit."

① Scheduling

Input: Collection of n jobs specified by their time intervals $[s_1, e_1], \dots, [s_n, e_n]$

Goal: Complete as many jobs as possible without any time conflicts

Example: [1,3], [10,11], [18,20], [2,19]

← do these 3

Greedy alg: Which job to pick next?

Pick the earliest end time? ✓

Pick the shortest!

X

Pick the earliest start time?

X

Claim 1: Greedy picks an optimal schedule $G = j_1 \cdots j_k$ (sorted by end time)

(Intuition) Suppose S is some optimal schedule

$[s_{i_1}, e_{i_1}]$ $[s_{i_2}, e_{i_2}]$

$S = i_1\ i_2 \cdots i_k\ i_{k+1} \cdots$ (sorted by end times)

$G = j_1\ j_2 \cdots j_k$

Suppose S and G agree on first m jobs

"Convert" S into S' which agrees w/ G on first $m+1$

Claim 2: $\forall\ m \le k$, $\exists$ opt sched S which agrees w/ G on first m jobs

Pf: By induction on m.

Base case: $m = 0$. ✓

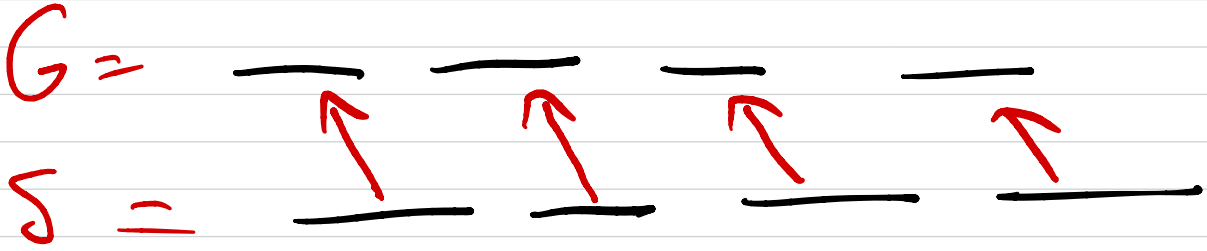
Inductive step: Suppose $i_1 = j_1, \ldots, i_m = j_m$

(i) $i_{m+1} = j_{m+1}$. ✓

(ii) $i_{m+1} \neq j_{m+1}$. Let $S' = S$ w/ i_{m+1} replaced by j_{m+1}.

Then S' is optimal, has no conflicts, agrees w/ G on first $m+1$. □

2 ⇒ 1: Suppose S is opt, agrees w/ G on first k. So $S = G$. □


G =
S =

Horn formulas

Variables: $x_1, \ldots, x_n \in \{\text{True}, \text{False}\}$ literal: x_i or $\overline{x_i}$

Clauses

1. Implications (with unnegated variables)

$(x_i \wedge x_j) \Rightarrow x_k$ $\quad\quad$ $\Rightarrow x_k$ $(x_k = \text{True})$

AND of any number of variables

2. Pure negative clauses

$(\overline{x_i} \vee \overline{x_j} \vee \overline{x_k})$ (OR of any number of negations)

Example: Dinner party among friends A, B, C, ..., Z. Who to invite?

$\Rightarrow C$ $\quad (A \wedge B) \Rightarrow I)$ $\quad (\overline{E} \vee \overline{F} \vee \overline{G})$

Horn-SAT

Input: Horn formula φ
Output: Satisfying assignment, if one exists

$\forall i$, set x_i = False

while some implication $(x_i \wedge \cdots \wedge x_j) \Rightarrow x_k$ is not satisfied
    Set x_k = True

if every pure negative clause $(\bar{x}_i \vee \cdots \vee \bar{x}_j)$ is satisfied
    Return $(x_1, \ldots, x_n)$

else
    Return "not satisfiable"

Example

Variables
x = ~~F~~ ~~T~~
y = ~~F~~ ~~T~~
z = F
w = ~~F~~ ~~T~~

Clauses

$(w \wedge y \wedge z) \Rightarrow x$ (✓)

$(x \wedge z) \Rightarrow w$ (✓)

$x \Rightarrow y$ ✓

$\Rightarrow x$ ✓

$(x \wedge y) \Rightarrow w$ ✓

$(\bar{w} \vee \bar{x} \vee \bar{y})$ ✗

Claim: (i) If Greedy outputs solution, then solution is satisfying.
(ii) If Greedy outputs "not satisfiable", the $\mathcal{X}$ is unsatisfiable.

Pf: (i) is clear ✓
For (ii), we'll prove the following...

Subclaim: At all steps in Greedy, if any set of vars is assigned True, then they are also True in every satisfying assign.

Pf: By induction on while loop:
Base case: all vars set to False. ✓

Induction step: Suppose $(x_i \wedge \cdots \wedge x_j) \Rightarrow x_k$ is selected.
Then LHS is true.
So LHS also true in every satisfying assignment.
Then $x_k = T$ in sat assgn. ∎

For (ii), assume false. Then Greedy outputs "no", but ∃ sat assgn.
⇒ Greedy violates $(\overline{x_i} \vee \cdots \vee \overline{x_j})$. Then sat assgn violates clause.
Contradiction! ◻

Data compression

We have an alphabet of 32 characters w/ frequencies $f_1, \ldots, f_{32}$.
Say we have a text with N characters and we want to encode it as efficiently as possible.

Naive: every character → 5 bits Overall: $5N$ bits

Freq	Example	Encoding #1	Cost	Encoding #2	Encoding #3	cost
0.4	A	00	$0.4N \cdot 2$	0	0	$0.4N \cdot 1$
0.2	B	01	$0.2N \cdot 2$	00	110	$0.2N \cdot 3$
0.3	C	10	$0.3N \cdot 2$	1	10	$0.3N \cdot 2$
0.1	D	11	$0.1N \cdot 2$	01	111	$0.1N \cdot 3$
			$2N$			$1.9N$

00|01|11|01

A B D B

000

AAA

AB B

BA

0|110|111|10

A B D C

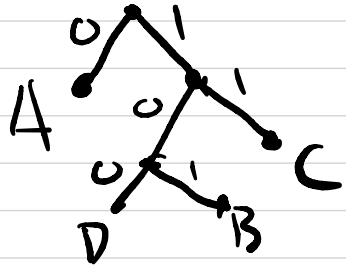
Prefix-free codes & trees

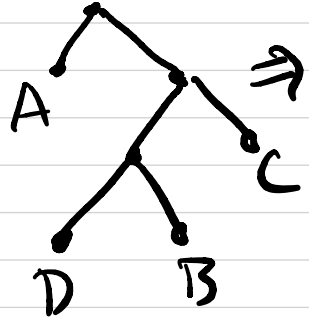
Any prefix-free code corresponds to a binary tree (and vice versa)

Example: A B C D

0 101 11 100

Prefix-free encoding:





A=0

B=101

for Horn-SAT

$(x \Rightarrow y)$ $(\bar{x} \vee y)$ same

"if x is True, then y is True"

x	y
T	T
F	T
F	F