daboost

Data: "Houston, we have a problem." **Pattern 1:** has **Pattern 2:** have

One By One

Search for has on all data - Failed

"Houston, we have a problem."

Search for have on all data - Success

"Houston, we have a problem."

Pros

✅ Simple to implement

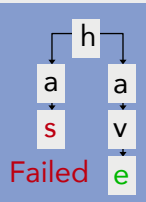
Cons

❌ Pass twice on all data

Decision Tree

Search for the letter h on all data

"Houston, we have a problem."



Compare each letter based on decision tree prepared

Pros

✅ Pass once on all data

Cons

❌ Complicated to implement tree

❌ Pre-process dictionary build

New - DaBoost.io Parallel Multi Pattern

Create dictionary based on used letters

"Houston, we have a problem."

	h	a	v	e
Value on 64bit register:	000	001	011	010
Add Value:	111	111	111	111
Compared to:	000	001	011	010 Success

Decimal value	Binary value	Letter
000	000	h
001	001	a
002	010	e
003	011	v
004	100	s
005	101	Any other letter

Pros

✅ Fast

✅ Pass once on all data

✅ Easy to use

✅ Text or Binary

Cons

❌ Pre-process dictionary build

DaBoost technology can be integrated in various applications, for example:

1. Different Parsers – HTML Parser, Office files parsers, Ethernet parser, 5G parser, Wi-Fi parser, BT5 parser
2. Distributed Non-Indexed NoSql Database query
3. Bio-Informatics: DNA sequencing, NGS
4. Cyber security: Firewalls, Anti-Virus, Hash pattern search

✅ Patent pending IL 281960

✅ Website: daboost.io