Sum of Subsets

using Backtracking

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array}$$
$\{10, 20, 30, 40\}$

sum = 50

$\{20, 30\}$

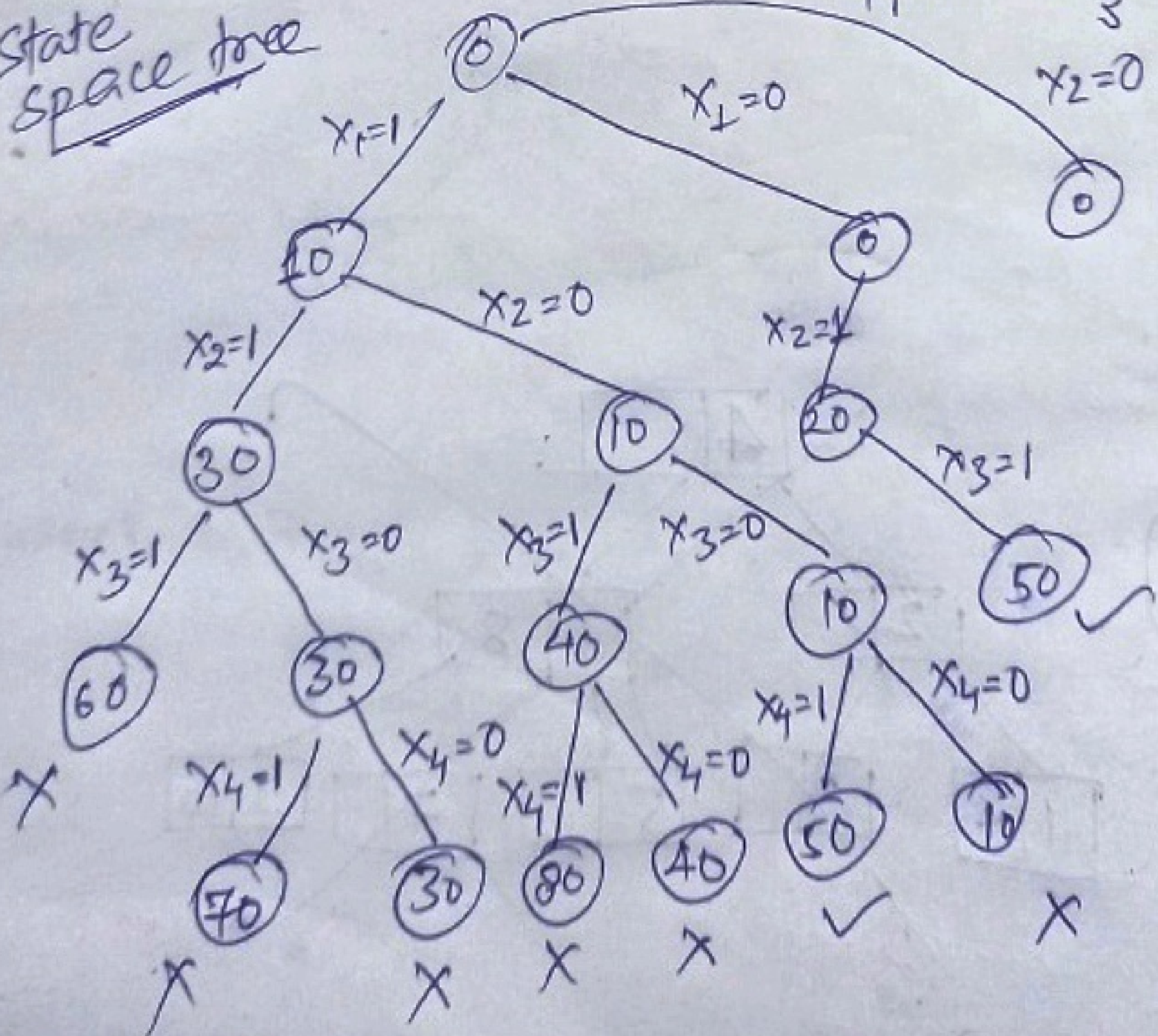
$\{10, 40\}$.

1	2	3	4
1	0	0	1
0	1	1	0

$2 \times 2 \times 2 \times 2 = 2^4$

$2 \times 2 \times \ldots 2 = 2^n$ { Bruteforce search ~~secre~~ approach }

State space tree



String Matching

→ find a pattern in a given text

text : a b c d a $\boxed{a\ b\ e}$ a b a c d ⇒ n

pattern : abe = m

$n \geq m$

1. Naive
2. Rabin-Karp

Naive String Matching Algorithm

pattern : abe

												No. of comparison ↓
Text :	a	b	c	d	a	b	f	a	b	e	g	
1st Itr	a	b	e									3
2nd Itr		a	b	e								1
3rd Itr			a	b	e							1
4th Itr				a	b	e						1
5th Itr					a	b	e					3
6th Itr						a	b	e				1
7th Itr							a	b	e			1
8th Itr ✓								a	b	e		3

No of comparision = 14.

Designing the Algorithm

input ⇒ Text & pattern
output ⇒ pattern found or not.

Naive string algo (T, P)

n = Text length

m = pattern length

for $s = 0$ to $n - m$
or $s = 1$ to $n - m + 1$

~~print("pattern if (a~~

if $P[1 \ldots m] = T[s+1, \ldots s+m]$

print ("pattern 'P' occurred at shift 's'");

∴ Time complexity $= (n-m+1)(m)$

$= nm - \cancel{m^2} + \cancel{m}$

$= nm$

∴ $\boxed{n > m}$ ∴ $O(nm)$

Bestcase :- $O(m)$

Rabin-karp Algorithm

↳ uses concept of hashing
↳ algo works on Numeric values

Given

Text : a b c d a b a b c d

pattern : $\boxed{daba}$ ⇒ hash function value

hash function ↳ Numerical

① ↳ $h(f) = \sum_{i=a}^{z} i$ summation of alphabets.

a = 1, b = 2, c = 3 ..

Ex ababc

1+2+1+2+3 = 9

Ex abbaab

1+2+2+1+1+2 = 9

② d d = 10 (value) × 10^{m-1}

Ex ababc m = 5

43210 $1\times10^4 + 2\times10^3 + 1\times10^2 + 2\times10^1 + 3\times10^0$

h(f) ⇒ 12123

Ex aabbba
543210

$1\times10^5 + 1\times10^4 + 2\times10^3 + 2\times10^2 +$
$2\times10^1 + 1\times10^0$

$h(f) = 112221$

Ex abcdeab
6543210

$1\times10^6 + 2\times10^5 + 3\times10^4 + 4\times10^3 + 5\times10^2$
$+ 1\times10^1 + 2\times10^0$

$h(f) = 1234512$

③ Modulus

$h(f) \Rightarrow$ By using modulus (q)
prime no.

$1234512 \bmod 11 = (\quad) < 11$
q

Base eq $\rightarrow$ $A - A \div B \times B$

$A \bmod B$.

Algorithm

① Calculate the hash values of pattern

② then we calculate sub(m) consecutive hash values

if $h(p) = h(\text{sub tex})$
then we move pattern comparison

~~text~~: a b b c a a c a

pattern : a c a

$h(f)$ = summation of alphabets
where a = 1, b = 2, ...

1st : a c a ⇒ 1+3+1 = 5
1+3+1

1st iter abb ⇒ 1+2+2 = 5
$h(p) = h(T_1)$ spurious hit
aca ≠ abb

2nd itr. bbc ⇒ 2+2+3 = 7
aca ≠ abb

3^{rd} Itr bca ⇒ 2+3+1 = 6 → spurious hit

4th Itr caa ⇒ 3+1+1 = 5   h(p) = h(T_4)   aca ≠ caa

5th Itr aac ⇒ 1+1+3 = 5   h(p) = h(T_5)   aca ≠ aac

6th Itr aca ⇒ 1+3+1 = 5

→ h(p) = h(T_6)   Actual hit

aca = aca

(2) d=10   value × $(d)^{m-1}$

Ex Text: a bb c aa ca

pattern: a c a   where a=1, 2.,
         2 1 0

h(p) = aca ⇒ $1 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$

⇒ 131

T_1 = abb = $1 \times 10^2 + 2 \times 10^1 + 2 \times 10^0 = 122$

T_2 = bbc = $2 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 223$

T_3 = bca = $2 \times 10^2 + 3 \times 10 + 1 \times 10^0 = 231$

T_4 = caa = $3 \times 10^2 + 1 \times 10 + 1 \times 10^0 = 311$

T_5 = aac = $1 \times 10^2 + 1 \times 10 + 3 = 113$

T_6 = aca = $1 \times 10^2 + 3 \times 10 + 1 = 131$

- there is not spurious hit

h(p) = h(T_6)   Actual Hit

aca = aca

ex aab = 122
bbc = $\boxed{223}$

$$\left[\text{hash value of previous } m \text{ character} - \left(\begin{array}{c}\text{value of}\\ 1^{st}\text{ character}\\ \text{of previous}\\ \text{iteration}\end{array}\right) \times 10^{(m-1)}\right] \times d$$

$$+ \begin{array}{c}\text{value of}\\ \text{last character}\\ \text{of existing iteration}\end{array}$$

$$(122 - 1 \times 10^2) \times 10 + 3 =$$

$$22 \times 10 + 3 \Rightarrow 223$$

aab = 122
bbc = 223
bca =

$$(223 - 2 \times 10^2) \times 10 + 1 = \boxed{231}$$

$$(223 - 200) \times 10 + 1$$

$$230 + 1$$

only for designing algorithm

③ Modulus

text: 3141592653589793

$m = 2$

pattern: 26 ; $q = 13$

$\Rightarrow h(P) = 26 \bmod 13 = 0$

$T_1 = 31$ $h(T_1)$ 31 mod 26 = 5

$T_2 = 14$ $h(T_2)$ 14 mod 26 = 14 = 1

$T_3 = 41$ $h(T_3)$ 41 mod 26 = 15 = 2

$T_4 = 15$ $h(T_4)$ 15 mod 26 = 15 = 2

$T_5 = 59$ $h(T_5)$ 59 mod 26 = 7

$T_6 = 92$ $h(T_6)$ 92 mod 26 = 14 = 1 Actual heit

$T_7 = 26$ $h(T_7)$ 26 mod 26 = 0

$h(p) = h(T_7)$ Spurious hit

26 = 26 $h(p) = h(T_8)$

65 mod 26 = 0 $26 \neq 65$

$T_8 = 65$ $h(T_8)$ 65 mod 26 = 0

$T_9 = 53$ $h(T_9)$ 53 mod 26 = 1

$T_{10} = 35$ $h(T_{10})$ 35 mod 26 = 9

$T_{11} = 58$ $h(T_{11})$ 58 mod 26 = 6

$T_{12} = 89$ $h(T_{12})$ 89 mod 26 = 11

$T_{13} = 97$ $h(T_{13})$ 97 mod 26 = 6

$T_{14} = 79$ $h(T_{14})$ $79 \bmod 26 = 1$

$T_{15} = 93$ $h(T_{15})$ $93 \bmod 26 = 2$

1 -> Spurious hit

1 -> Actual hit - 7th Itr

Rabin - Karp (T, P, d, q)

1. n = text length
2. m = pattern length
3. $h = d^{m-1} \bmod q$ → hash value calculation
4. $p = 0$ → initial value of pattern
5. $t_0 = 0$ → initial value of hash function
6. for 1 to m

   $p = (dp + P[i] \bmod q)$ → hash value of pattern

   $t_0 = (dt_0 + T[i] \bmod q)$ → hash value of first 'm' text characters

7. for s=0 to n-m

   if $p == t_s$ → if hash matches then go for string comparision

   if $P[1 \ldots m] = T[s+1 \ldots s+m]$

   print ["pattern found at shift s");

if $s < n-m$

$$t_{s+1} = \left(d\left(t_s - T(s-1)h\right) + T(s+m+1) \right) \bmod q.$$

↳ hash value of next character using the formula.

Time Complexity

$O((n-m+1)(m))$

$nm - \cancel{m^2} + \cancel{m} \quad n > m$

$O(nm)$