CSE221 ALGORITHMS

Topic: Greedy Methods (Huffman Coding, Fraction Knapsack)

Prepared by:
Saad Bin Sohan
BRAC University

Email: sohan.academics@gmail.com

GitHub: https://github.com/saad-bin-sohan

Greedy Methods

→ it considers the best (most benificial) option available right now (without thinking about the future)

e.g: Algorithms to find minimum spanning tree ⇒ Kruskal Algo. Prim algo (Finds the currently available least weighted edge).

Dijkstra algorithm is also greedy method approach

Huffman Coding

→ Huffman encoding and decoding both করতে।

Encoding:

Message / data ——converts to——→ bit (0's and 1's)

Decoding:

Encrypted bits ——converts to——→ Message

prior to Huffman encryption এর জন্য each characters is represented in 8 bits using ASCII

∴ Lana Del Rey এই word টা pc represent করতে 12 × 8 = 96 bits নিবে।

- 12 → 12 characters (including two spaces)
- 8 → bits to represent each characters

এই conversion এর জন্য Huffman efficient process টি হলো।

NB: Huffman method এ each character (including symbols, punctuations, spaces etc) is treated as a unique entity.

∴ A ≠ a

e.g. Huffman encoding process :-

step 1) count repetition of each characters

step 2) Merge the two characters with least count and keep generating a tree (and discard those two characters

NB: You will start from the bottom in generating that tree. if more than two characters (or character set) has that same least count, you can choose randomly.

e.g. of Huffman Encoding:

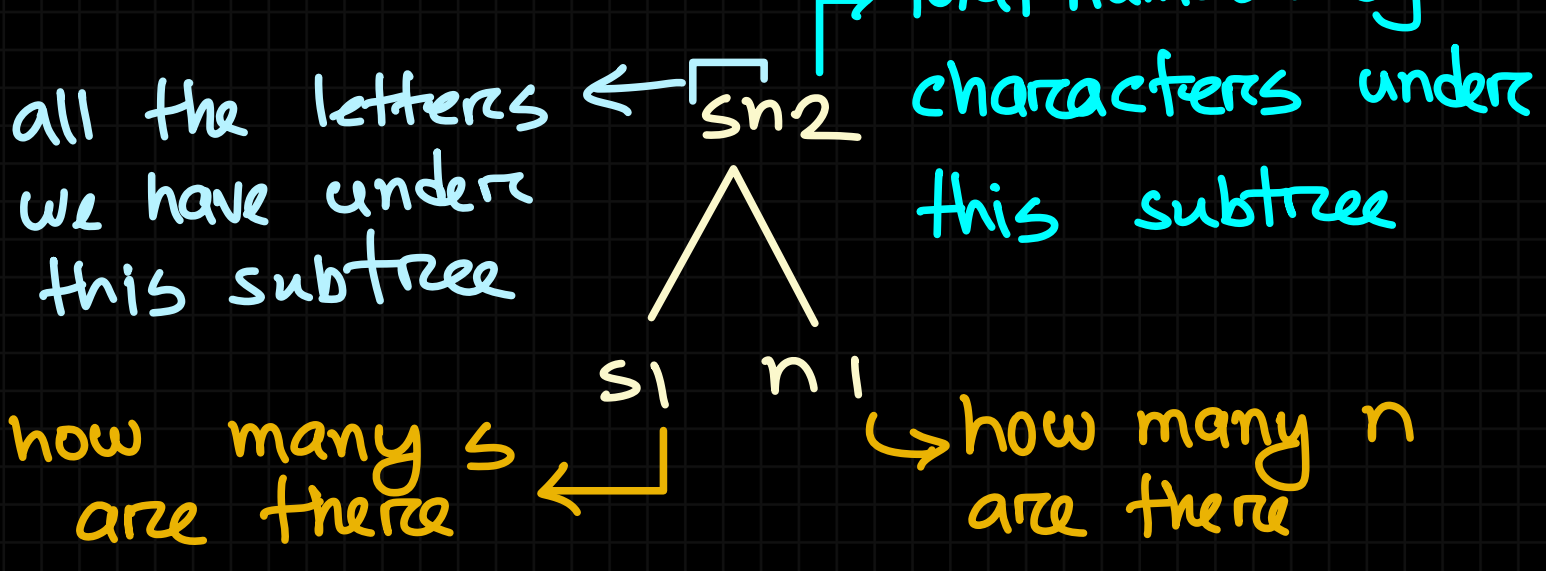
example of message = opposition

step-1:

o → 3
p → 2
i → 2
t → 1
s → 1
n → 1

step-2:

take n and s out of n, s, t



after taking s1 and n1 into the tree we remove them from the available letters from tree and we keep repeating this process until no character (or character set) is left as available option.

now.

O → 3

P → 2

i → 2

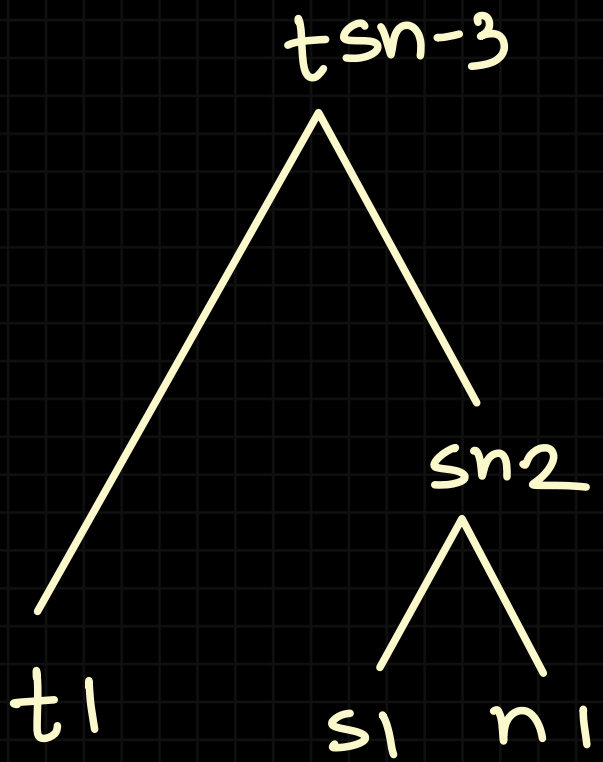
t → 1

~~s → 1~~

~~n → 1~~

sn → 2

now take sn2 and t1



0 → 3

p → 2

i → 2

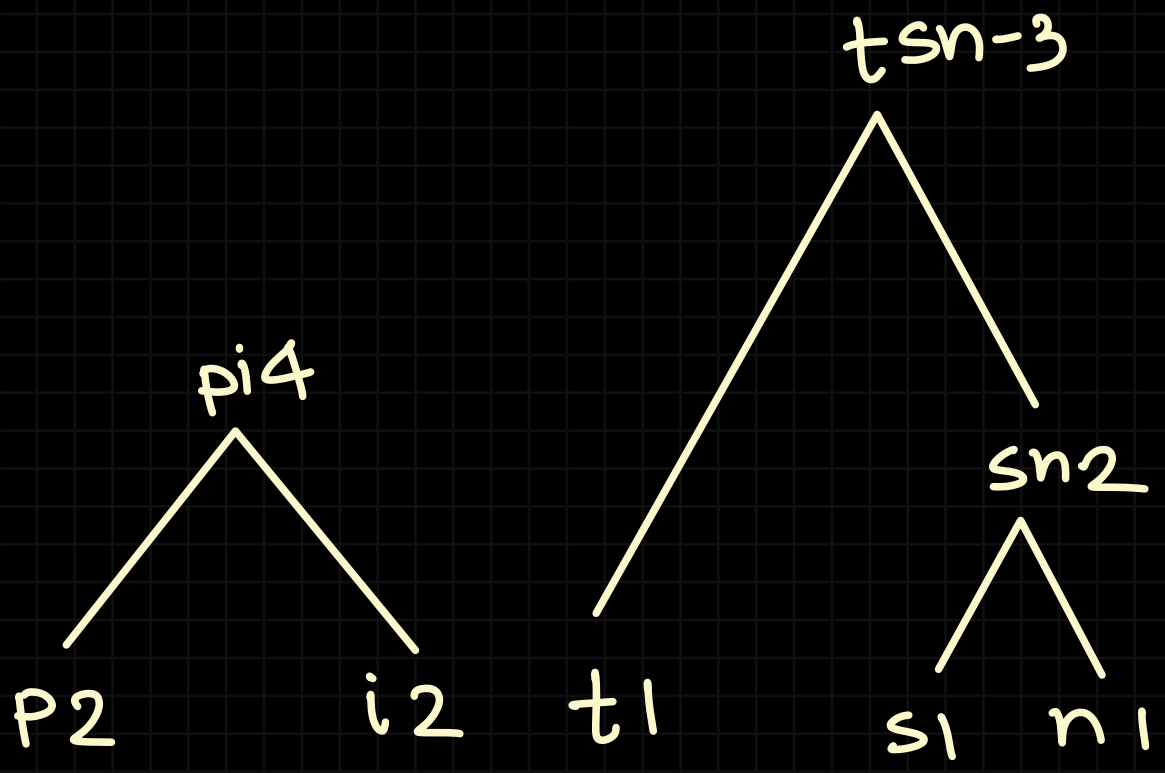
~~t → T~~

~~s → T~~

~~n → T~~

~~sn → 2~~

tsn-3



0 → 3

~~p → 2~~

~~i → 2~~

~~t → 1~~

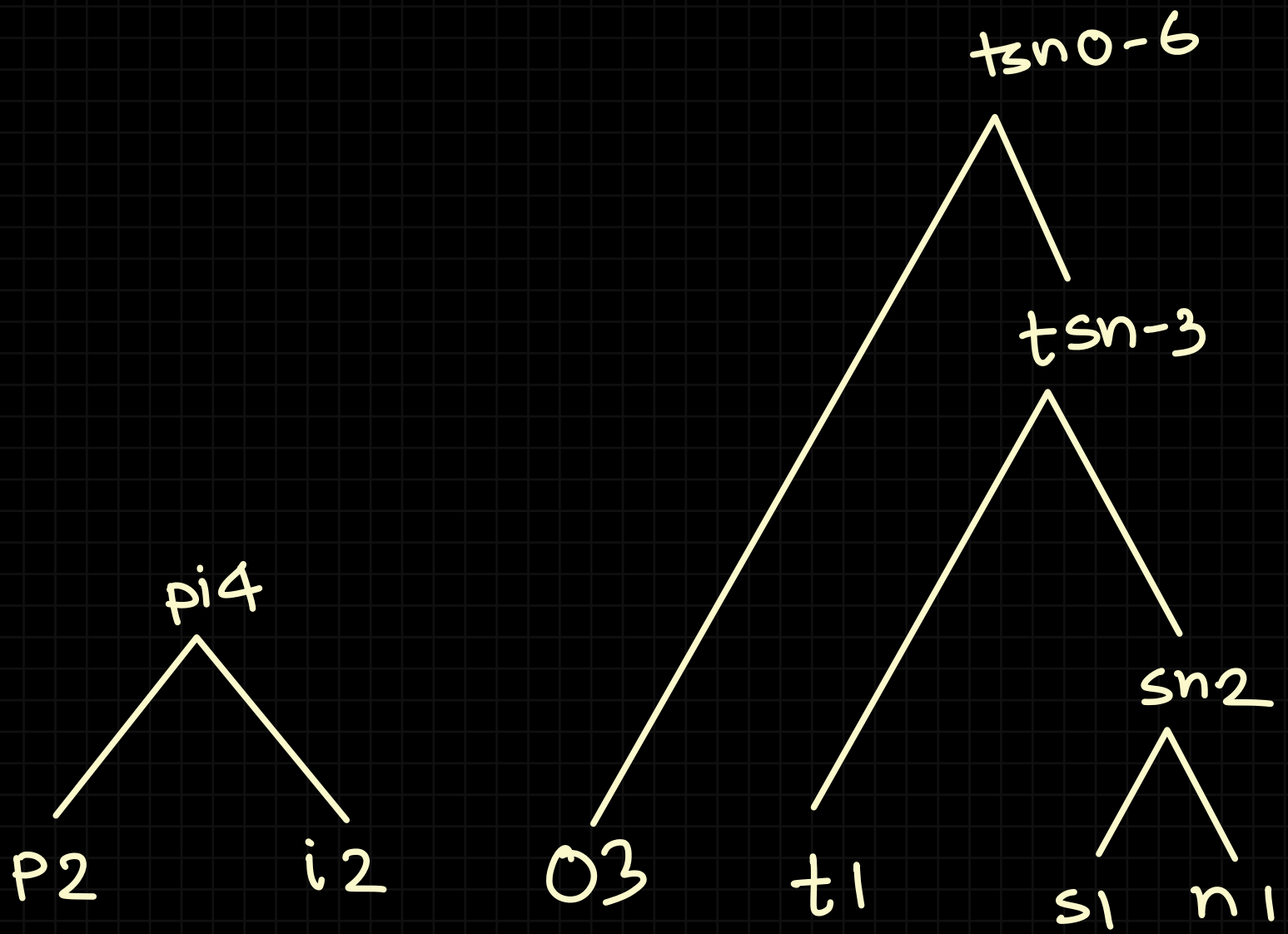
~~s → 1~~

~~n → 1~~

~~sn → 2~~

tsn-3

pi → 4


tsn0-6
tsn-3
pi4
sn2
p2
i2
03
t1
s1
n1

o->3

p->2

i->2

t->1

s->1

n->1

sn->2

tsn-3

pi->4

tsno-6

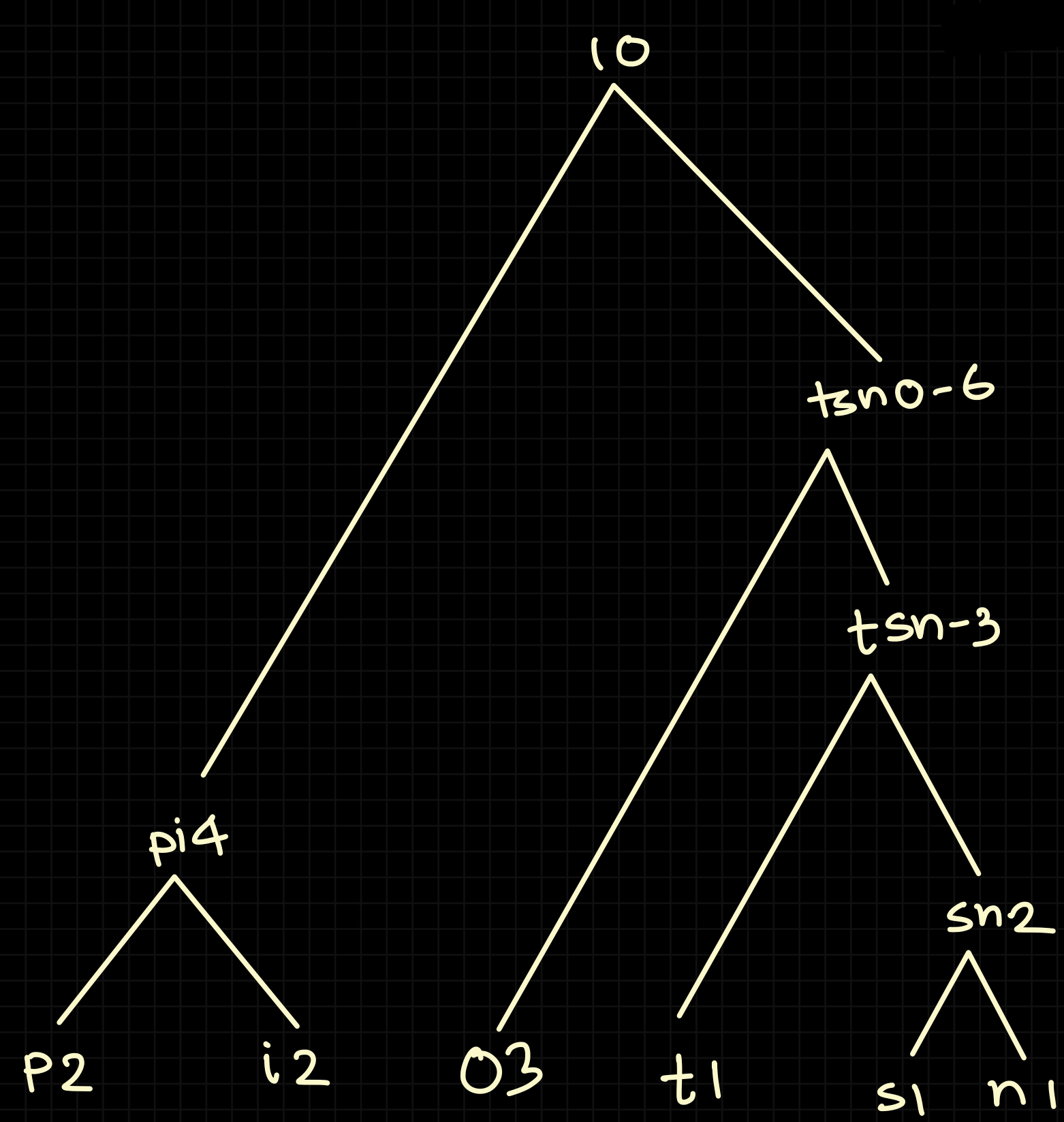
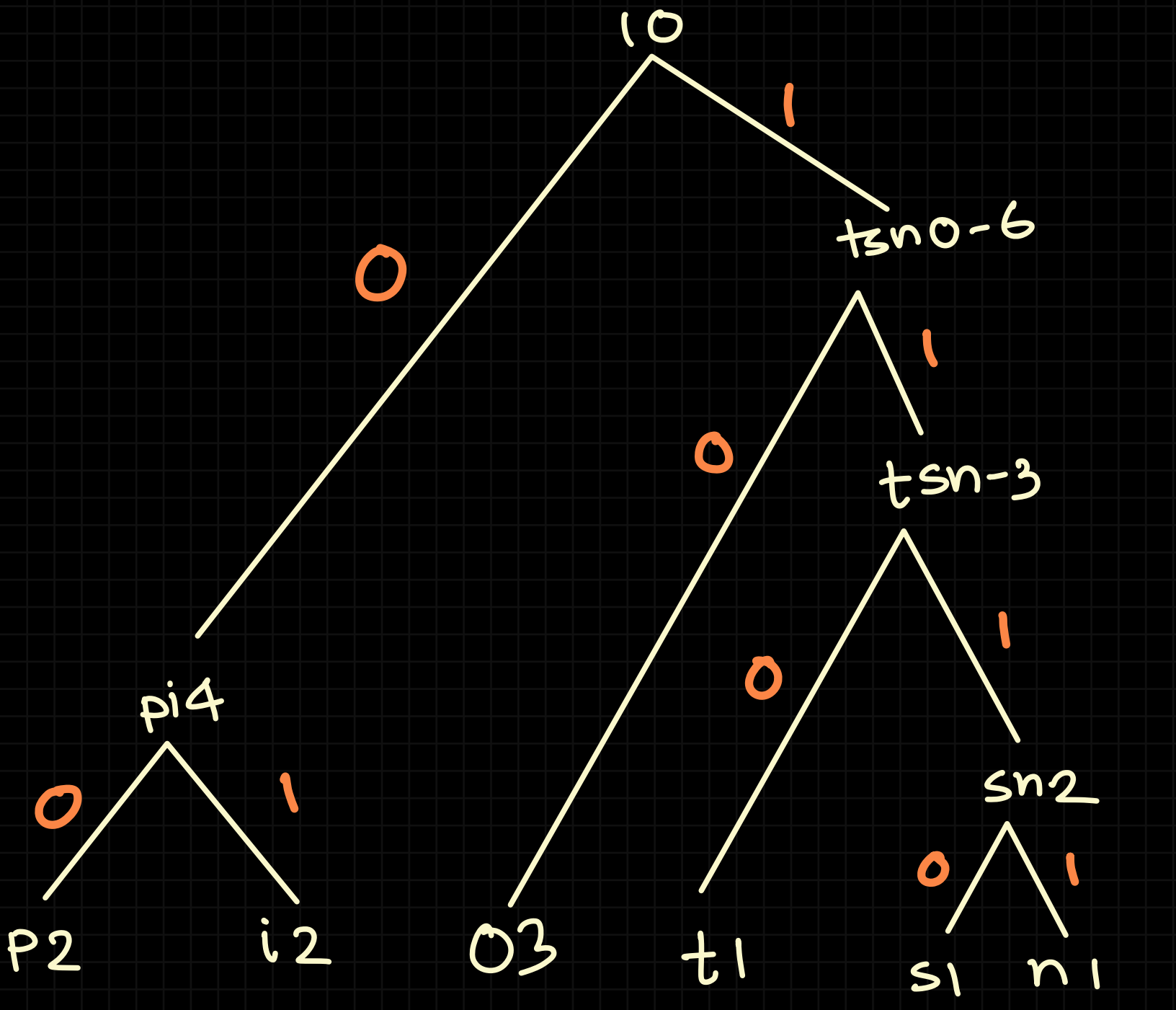


figure: Huffman Tree

now, left edge → 0
right edge → 1

now, halfman tree ? starting from the top, প্রতি subpart (or level or edge) এর জন্য left part কে '0' and right part কে '1' হিসেবে mark করি।



now the query is huffman algorithm এ কীভাবে কোনো letter (e.g. P) কে কোন্ bit দিয়ে represent করবো?

ans: root থেকে start করে target letter (P) এ যেতে হবে। যেতে যেতে যে edge value পাবো thats the representation of that letter.

so, for P → 0 0

root থেকে 0 bit then 0 bit cross করলে P পাওয়া যাচ্ছে।

∴ representation of each letter:

0 → (10)

P → (00)

s → (1110)

i → (01)

t → (110)

n → (1111)

using this bit representation we reach our target of representing all the letters (in the data / message) using **minimum number of bits.**

∴ now,

O(10)P(00)P(00)O(10)S(1110)i(01)t(110)i(01)O(10)n(1111)

∴ opposition in huffman coding is

opposition = 1000 0010 1110 01110 01101111

total 25 bits

ASCII use করতে লাগবে → 10*8=80 bits

Huffman encoding use করে → 25 bits

speed, space efficiency

in those binary bits notice, যে letter বেশি তার representating bit কম (e.g. 0 ৳ 3 bit) and কম count এর letter যে বেশি (৸ 4 bit ৺).

so বেশি count এর letter গুলো কম bit (space বেচে)

Huffman Decoding

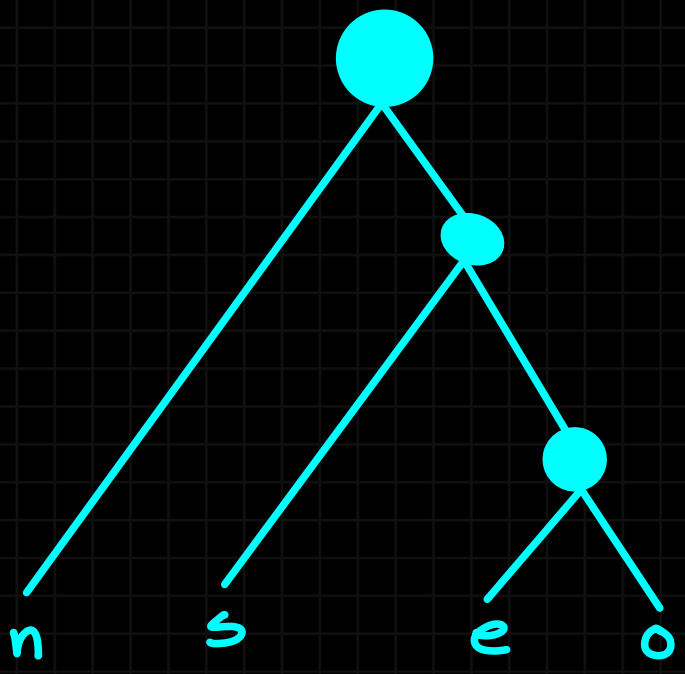
decoding = encoded bits কে message এ decrypt করা

→ Huffman tree and bits দেওয়া থাকবে। Decode করে message বের করা লাগবে।

Que Bits:

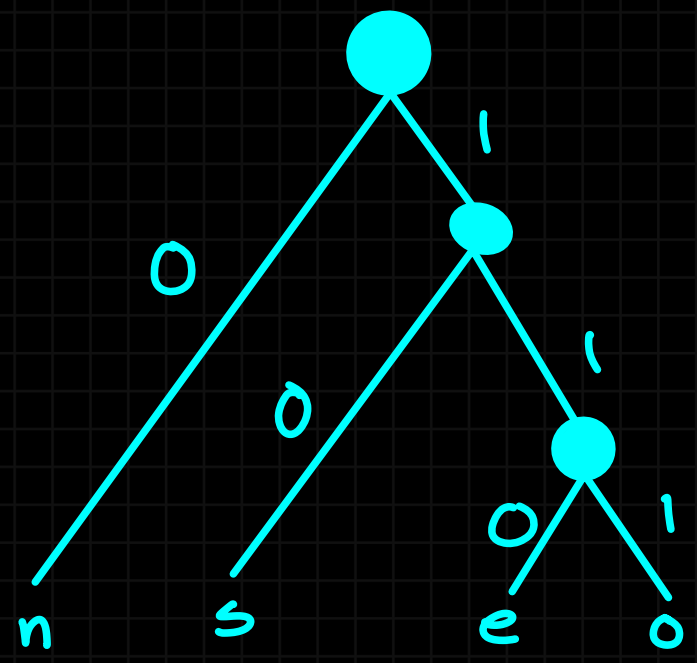
0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 0

Huffman tree:



message টা কে decode করো।

soln: We write edge bits



0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 0

n o n s e n s e

process: each time given bit sequence এর unreached bit থেকে start করে bit sequence follow করে root থেকে শুরু করে আসতে until you reach the last level (or a letter)

decoded message: nonsense

Fractional Knapsack

imagine you are a thief. Problem is when stealing someone, you have a knapsack (অস্ত্র) that a limitation for what it can carry. So your target is to steal the right amount of the right stuffs so you get the maximum value for their market price.

Good thing: As the name suggests

you can choose partial portion amount of an item available

e.g. You can steal 2.5 kg of 5kg available gold.

You can also not select a particular item also. But then you won't get any value for that.

Que knapsack = 5 kg

item	price	weight (available amount)
A	140	2
B	200	1
C	150	5
D	240	3

কোন item কী পরিমাণ নিলে max benefit হবে?

(think before seeing the answer)

soln: start with the item that gives maximum worth of value for per unit and take maximum carryable amount of it and for the rest available space in knapsack,

item	price	weight	price per weight
A	140	2	70
B	200	1	200
C	150	5	30
D	240	3	80

choose the next item with most value per item and choose max possible amount of that. Repeat the process until knapsack is full.

so

B (1 unit) + D (3 unit) + A (1 unit)

= B (200) + D (240) + A (70)

= 510